NOVA IMS Information Management School

MGI

Mestrado em Gestão de Informação
Master Program in Information Management

Uma nova abordagem de Big Data através da implementação do Software RAID

Catarina Jacinto Nunes da Costa Amorim

Trabalho de Projeto apresentado como requisito parcial para obtenção do grau de Mestre em Gestão de Informação, Especialização em Gestão do Conhecimento e Business Intelligence

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

UMA NOVA ABORDAGEM DE BIG DATA ATRAVÉS DA IMPLEMENTAÇÃO DO SOFTWARE RAID

por

Catarina Jacinto Nunes da Costa Amorim

Trabalho de Projeto apresentado como requisito parcial para a obtenção do grau de Mestre em Gestão de Informação, Especialização em Gestão do Conhecimento e Business Intelligence

Orientador: Professor Doutor Roberto Henriques

Fevereiro 2019

AGRADECIMENTOS

Sem dúvida uma das partes mais difíceis de escrever com tantas pessoas para agradecer e tão pouco espaço para o efeito. Para começar, gostaria de agradecer a todas as pessoas que me acompanharam ao longo desde processo que me ajudaram e contribuíram para a existência deste projeto.

Aos meus pais, Luís e Fátima, e irmão, Tiago, um obrigado do tamanho do mundo pelo apoio, paciência e carinho. Pelo exemplo de perseverança que me foram passando ao longo da minha vida académica. Mas acima de tudo todo o esforço feito ao longo desta caminhada para tornar os meus sonhos possíveis.

Ao meu namorado, Miguel, que nunca me deixou desistir, pelo apoio incondicional, sentido crítico e compreensão. Pedir desculpas pelos vários programas que tivemos de abdicar para que este trabalho fosse desenvolvido.

À Tia Otília que sempre que haviam eventos familiares para além de me "dar na cabeça" por este projeto ainda não estar acabado, dava-me sempre a motivação extra para conseguir. Tia, agora já te posso responder "Sim, já está acabado e entregue!"

Ao Professor Doutor Roberto Henriques que esteve presente desde o início do meu percurso académico e pelo seu esforço e dedicação ao rever este relatório.

À WeDo Technologies pela oportunidade e disponibilidade de poder desenvolver este projeto. Não obstante ao papel fundamental que o Daniel Castelhano, Marco Angélico e Ricardo Murta tiveram dentro da empresa ao proporcionarem a informação e conhecimento necessários.

RESUMO

Vivemos num mundo cada vez mais competitivo, onde conhecer quais são as nossas vantagens competitivas torna-se algo fundamental. Assim, as organizações pretendem obter um acesso rápido e eficiente aos seus dados e à informação que é possível produzir através dos mesmos. Este aspeto nos dias que correm, com o aumento dos dados, torna-se cada vez mais difícil e é essencial para as organizações manterem a sua estratégia e a estrutura.

Posto isto, o setor das telecomunicações enfrenta um paradigma de sobreprodução de dados nas suas atividades diárias com tendência a aumentar.

O objetivo deste projeto é a implementação de uma nova tecnologia de suporte ao software RAID. Esta nova abordagem é alicerçada em Big data que pretende substituir o suporte atualmente realizado em base de dados tradicionais. Para o efeito testa-se a rapidez na leitura dos dados em RAID suportado por tecnologia Big Data, em comparação com a rapidez da solução atual que passa pela leitura dos dados suportada por RBDMS.

PALAVRAS-CHAVE

Big Data; RAID; Hadoop; WeDo Technologies; Warehousing; Performance

ÍNDICE

ÍNDICE ILUSTRATIVO

ÍNDICE DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

TI	Tecnologias de Informação
BI	Business Intelligence
DSR	Design Science Research
ETL	Extract Transform Load
GB	Gigabyte
EB	Exabyte
CDR	Call Detail Records
CRM	Customer Relationship Management
ERP	Enterprise Resource Management
TB	Terabyte
HDFS	Hadoop Distributed Filesystem
DW	Data Warehouse
PMS	Performance Measurement System
RDBMS	Relational DataBase Management System
OLAP	on-line analytical processing
OLTP	on-line transaction processing
BPM	Business Process Management

1. INTRODUÇÃO

Atualmente temos testemunhado um crescimento massivo da quantidade de dados existente no mundo. Tendo o Big Data chegado a um ponto sem retorno nos diferentes setores da economia mundial e o rápido desenvolvimento das tecnologias informação têm intensificado o seu crescimento. Mas a principal questão que se coloca é qual o impacto que esse crescimento na quantidade de dados pode ter no paradigma atual e futuro. Muitos consumidores desconfiam da quantidade de dados recolhidos sobre os mais variados aspetos das suas vidas, desde os seus hábitos de consumo até ao seu estado de saúde. O que nos levanta questões de ordem ética e moral. É imperativo colocar-se a seguinte questão: "Is big data simply a sign of how intrusive society has become, or can big data, in fact, play a useful role in economic terms that can benefit all societal stakeholders?" (McKinsey & Company, 2011)

Estima-se que desde 2005 até 2020 o universo digital irá expandir-se em 300%. Consequentemente, o investimento das empresas em infraestruturas como Hardwares, Softwares, serviços, telecomunicações e mão de obra irá crescer cerca de 40%. Aliado a isso, terão de ser considerados outros investimentos em áreas como gestão de armazenamento, a segurança, Big Data e Cloud que irão crescer consideravelmente rápido. A IDC prevê que em 2020 cerca de 33% do universo digital trará informação que possa ter valor para ser analisada (Gantz & Reinsel, 2012).

Para suportar esse crescimento surgiram alguns softwares capazes de armazenar essa enorme quantidade de dados, como por exemplo o Hadoop. Uma tecnologia *Open Source* em Java que ajuda a armazenar, aceder e a obter grandes porções de informação a partir do Big Data de maneira distribuída a um menor custo, com alto grau de tolerância a falhas e alta escalabilidade (Saraladevi, Pazhaniraja, Paul, Basha, & Dhavachelvan, 2015)

A WeDo Technologies como empresa especializada na área da produção de software das telecomunicações tem se empenhado de forma bastante ativa na procura de soluções para este novo paradigma. O RAID, software produzido pela WeDo Technologies, permite simplificar as Tecnologias de Informação (TI) numa única plataforma de modo a recolher, transformar, armazenar e analisar de modo detalhado dados para dar suporte ao negócio em tempo real (W. Technologies, 2018).

Posto isto, este projeto foi planeado com o intuito de testar uma nova solução capaz de melhorar a performance do software RAID, de modo a apresentar novas soluções às empresas clientes da WeDo Technologies através do fornecimento de um software de integração de dados ajudando a tomarem melhores decisões de modo a gerir e otimizar o negócio.

A solução atual está alicerçada num processo tradicional de Data Warehousing (base de dados especializadas otimizadas para análises e frequentemente usadas para armazenar grandes quantidades de dados estruturados onde os dados são carregados através de processos ETL (McKinsey & Company, 2011)). Pretende-se testar uma solução orientada para a tecnologia de Big Data adotando uma abordagem flexível e multidisciplinar (McKinsey & Company, 2011).

Desta forma o armazenamento e leitura dos dados terá de ser revista e ajustada a esta nova realidade. Inicialmente, os dados a serem utilizados para a realização deste projeto seriam fornecidos pela a WeDo Technologies. Contudo, esse facto não se sucedeu devido à nova política de proteção de dados

e às repercussões que isso poderia ter para a empresa. Para contornar esta situação foram utilizados dados de Open Data.

Serão realizados três momentos de análise ao longo do projeto. Numa primeira fase, realizar-se-á uma análise dos dados de modo a perceber se existe a necessidade de proceder a alguns processos de limpeza de dados. Numa segunda fase, será desenvolvida uma solução utilizando a tecnologia de Big Data. Numa terceira fase, irá se proceder à avaliação da performance global da solução tendo em conta a tecnologia desenvolvida no ponto anterior. Em concreto, avaliar-se-á a eficiência do mesmo através da comparação de tempos de processamento entre a abordagem tradicional e esta nova solução proposta. Outro aspeto que terá de ser tido em conta prende-se com a necessidade de alocação de recursos na criação de conhecimento da ferramenta por parte dos colaboradores que irão desenvolver os projetos.

1.1. CONTEXTUALIZAÇÃO

Hoje em dia, o volume de criação dos dados em tempo real cresceu de GB em 2005 para EB em 2015 (Saraladevi et al., 2015) marco esse que deve ter sido largamente ultrapassado nos dias de hoje. De modo a ter uma boa implementação de BI são necessários sistemas de implementação técnicos, negócio significante ou domínio do conhecimento assim como uma capacidade de comunicação efetiva (Chen & Storey, 2012) . Tendo em conta esta evolução gigantesca nos dados surgiu o conceito de Big Data.

Big Data refere-se a uma enorme quantidade de dados, quer sejam eles estruturados ou não estruturados e às ferramentas capazes de armazenar, recolher e transformar esses mesmos dados. Caracteriza-se, ainda, pelos 4V's: Volume, Variedade, Velocidade e Variabilidade (Dijcks, 2012).

"Mas o importante não é a quantidade de dados. E sim o que as empresas fazem com os dados que realmente importam" (SAS, n.d.).

Na área das Telecomunicações, nomeadamente na Índia, as redes desde a mediação até à faturação tipicamente geram de 100 milhões a meio bilião de CDRs por dia. Posto isto, os técnicos devem usar esses dados para procederem a transações, monitorizar serviços e atividades de faturação, gerir vendas e preparar iniciativas de Marketing (Kumar, 2012).

Para suportar as grandes quantidades de dados gerados surgiu o Hadoop. Que é uma "open source (free) software framework for processing huge datasets on certain kinds of problems on a distributed system" (McKinsey & Company, 2011).

O Hadoop é a implementação mais popular e open source de MapReduce, sendo considerado uma estrutura de software ou programação de modelos confiável, escalável, paralela e distribuída. Em vez de softwares e sistemas caros para processar e armazenar os dados, o Apache Hadoop permite o processamento paralelo de grandes dados em cluster (Ghazi & Gangodkar, 2015).

A WeDo Technologies é uma empresa líder mundial na área do Revenue Assurance e Fraud Management pertencente ao grupo Sonae, mais particularmente à SonaeIM. Foi fundada em 2001 tendo como cofundador e CEO Rui Paiva, tendo completado recentemente 16 anos de existência.

Inicialmente focou a sua atividade no mercado português, mas rapidamente a potencialidade do produto foi constatada fora de portas e a empresa procedeu à sua internacionalização.

O RAID é o principal software desenvolvido pela WeDo Technologies. Consiste numa ferramenta capaz de recolher, analisar e correlacionar os dados de modo a traduzi-los em informação ajudando na tomada de decisão no negócio dos clientes em tempo real. Assumindo, assim, uma abordagem tradicional de uma arquitetura de BI (W. Technologies, 2018).

O objetivo é os clientes aumentarem o seu valor e as suas vantagens competitivas com um software rápido e user-friendly onde a monotorização do negócio seja fácil de manobrar e com todas as ferramentas necessárias para manipular e recolher os dados num só software.

Atualmente, um dos constrangimentos que enfrenta prende-se com a lotação dos servidores que com o avanço dos projetos vão ficando com menos espaço e capacidade de resposta. Através de uma plataforma Hadoop, seria possível gerir de forma mais eficiente a capacidade e processamento dos servidores agilizando todo o processo. Esta solução traria vantagens em toda a cadeia, isto é, desde a parte do desenvolvimento que, teoricamente, se tornaria mais rápido até à implementação do produto e utilização por parte do cliente.

1.2. Identificação do Problema

Este projeto pretende perceber de que modo é que esta nova abordagem ajudará as empresas da área das telecomunicações a lidar com a imensa quantidade de dados existente decorrente da sua atividade de negócio.

Posto isso, proponho responder às seguintes questões:

1. Porque que as aplicações de Big Data são o futuro das empresas de telecomunicação?
2. De que modo esta nova abordagem melhora a performance do programa?
3. Que impacto irá ter nos clientes da WeDo Technologies?
4. Que alterações ao processo de negócio da empresa terão de ser feitas?

1.3. Relevância e Importância do Estudo

Considerando a quantidade de dados que está a ser gerada e que se prevê que seja produzida até 2020, um conjunto variado de técnicas e tecnologias foram desenvolvidas e adaptadas para agregar, manipular, analisar e visualizar Big Data (McKinsey & Company, 2011). Aceitando esse facto, o maior objetivo deste projeto será provar que mudando a abordagem de armazenamento e processamento dos dados para novas tecnologias, poderá provar-se a existência de uma inovadora e mais apropriada abordagem.

Tendo em conta o paradigma que se vive sobre o grande volume de dados, os clientes têm solicitado com maior frequência uma solução mais robusta para lidar com esta realidade. As tecnologias de Big Data poderão vir a melhorar a performance de um software ligado ao mundo das telecomunicações.

Visto que a WeDo Technologies começa a ter os seus processos de ETL obsoletos em servidores onde se encontram mais que um projeto, é fundamental comparar o método atual com um método onde o RAID assenta sobre a infraestrutura Hadoop e de que forma ela poderá ajudar a eliminar a lacuna existente neste aspeto. Aliado a isso, alguns clientes da empresa já propuseram o uso deste método.

De modo a aproveitar o melhor do Big Data as empresas devem envolver as suas infraestruturas de TI para suportar os novos grande volumes, velocidades e variedades de dados e integra-los com os existentes dados empresariais a serem analisados (Dijcks, 2012). Deste modo, é fundamental incutir os conceitos de Big Data na organização, de maneira a não perder as suas vantagens competitivas no mercado. Posto isto, o projeto irá usar como ferramentas principais o RAID, Oracle e o Hadoop, mais propriamente o HDFS, Hive e Impala.

1.4. OBJETIVO DO ESTUDO

Este projeto tem como objetivo criar uma nova abordagem de modo a apresentar as diferenças entre a utilização de um software que utiliza uma arquitetura tradicional de BI e a utilização de ferramentas de Big Data. Pretende-se estudar como esta abordagem afeta a performance do software.

Os testes feitos ao longo do projeto serão úteis para mim a nível académico e a nível profissional. A WeDo Technologies também poderá tirar proveito do trabalho desenvolvido para encarar os desafios atuais.

A arquitetura proposta deve cobrir testes feitos através do armazenamento dos dados em HDFS e leitura dos dados através do Hive e Impala, comparativamente à solução existente que retém os dados em servidores Linux e faz a leitura dos dados através de Base de Dados Oracle.

Para atingir esse propósito são apresentados nos pontos a baixo os objetivos a desenvolver para desenhar a arquitetura desejável.

a) Identificar os fatores comuns e diferenças a serem considerandos entre ambas as abordagens. Qual será a influência e impactos que terá para a empresa.

b) Estudo das melhores práticas a serem tomadas em ambas as abordagens.

c) Desenvolvimento das diferentes abordagens tendo em conta os aspetos anteriormente referidos.

d) Avaliação dos métodos, procurando perceber qual dos métodos será uma mais-valia para a empresa.

1.5. ESTRUTURA

Relativamente à estrutura a ser usada no presente relatório, este projeto está assente em 5 capítulos:

1. **Introdução:** Além de uma contextualização do paradigma atual das tecnologias de Big Data, é descrito o objetivo do estudo e as questões à qual o mesmo pretende responder.

2. **Revisão de Literatura:** Consiste na realização de uma pesquisa detalhada de aplicações na área de Big Data, de forma a averiguar o que terá sido feito até agora e qual o ponto de partida para o futuro. Realizar-se-á a leitura de material especializado através de artigos académicos, participação em conferências do tema e estudo de material diretamente fornecido pela empresa.

3. **Metodologia:** Irá proceder-se à recolha de informação com programadores base do software RAID para avaliar a viabilidade do estudo. Todo este processo será consolidado com a experiência de trabalho na empresa que permitirá contactar com diferentes realidades de forma a poder desenvolver novas soluções. A conceptualização do artefacto terá por base os elementos recolhidos na revisão de literatura. Nesta fase será feita também a recolha de dados para o estudo e a operacionalização do artefacto conceptualizado. Por fim, irá se proceder-se ao teste da solução.

4. **Resultados:** Serão analisados e avaliados os resultados tendo em conta a metodologia acima citada.

5. **Conclusões:** Irá conter as conclusões e as limitações que condicionaram o projeto. Será feita uma reflexão acerca do cumprimento dos objetivos indicados no capítulo anterior e que trabalhos poderão ser efetuados no futuro.

2. REVISÃO DE LITERATURA

No presente capítulo serão apresentadas as bases teóricas que servirão para o desenvolvimento deste projeto, contendo diversas definições de Business Intelligence e Big Data. Assim como a arquitetura de algumas aplicações de Big Data a serem usadas como o HDFS, Hive e Impala.

Contém ainda, o enquadramento da empresa WeDo Technologies no mercado Tecnológico português e internacional e do seu software de Business Intelligence RAID.

2.1. BUSINESS INTELLIGENCE

O termo Business Intelligence foi introduzido pelo grupo Gartner em 1996, referindo-se a uma ferramenta e tecnologias como o Data Warehouse, relatórios e análises. Consiste num processo de recolha, armazenamento, tratamento e difusão de informação de modo a ajudar as empresas na tomada de decisão. Considera-se que um bom sistema de Business Intelligence fornece boas capacidades de perceção ao utilizador do que se está a passar dentro da organização. Tem como objetivo criar respostas e analisar novas questões (Stage, 2006).

O relatório da McKinsey Global Institute define BI como "A type of application software designed to report, analyze, and present data. BI tools are often used to read data that have been previously stored in a data warehouse or data mart. BI tools can also be used to create standard reports that are generated on a periodic basis, or to display information on real-time management dashboards" (McKinsey & Company, 2011).

Para Dayal, "Business Intelligence involves the integration of core information with meaningful business information to detect significant events, discover new business scenarios and predict business situations. It includes the ability to monitor business trends, to evolve and adapt quickly as situations change and to make intelligent business decisions on uncertain judgments and contradictory information." BI é assim uma ferramenta que transforma dados em conhecimento, que auxilia os gestores na tomada de decisão. As soluções de BI são uma realidade bastante presente na vida das empresas ligadas ao ramo das telecomunicações, devido à enorme quantidade de dados que estas produzem. Vercellis corrobora esta afirmação indicando que (Vercellis, 2009) "BI plays a significant role in the telecommunication industry due to the availability of large volume of data and the rigorous competition in the sector" (Castellanosm M., 2008; Olaru, 2014).

Segundo Negash, o BI combina dados operacionais como ferramentas analíticas. Refere, ainda que o objetivo é obter a informação o mais instantânea possível e melhorar a qualidade dos dados para os tomadores de decisão (Negash, 2004).

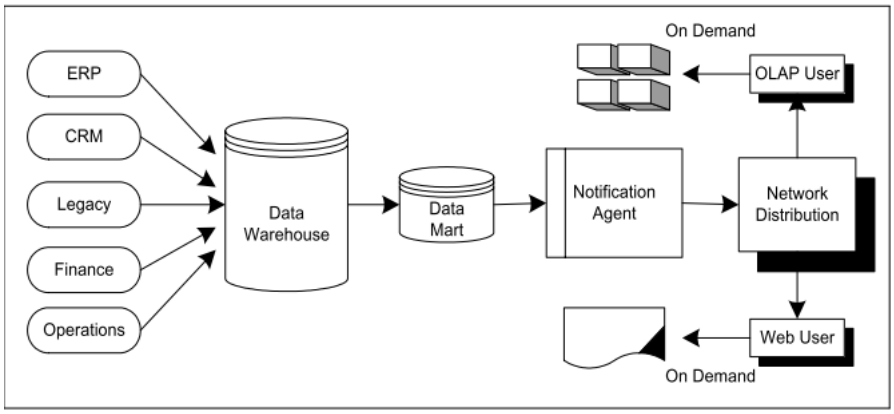


Figura 2.1 - Arquitetura típica de BI (Negash, 2004)

Sendo os Data Warehouse são bases de dados otimizadas para relatórios e frequentemente usadas para armazenar grandes quantidades de dados estruturados. Os dados são carregados através de um processo de ETL, que é uma ferramenta de software usada para extrair dados de fontes externas, transforma-los e aloca-los para necessidades operacionais e carrega-los numa base de dados ou num DW de dados de fontes operacionais para a elaboração de relatórios. Estes são geralmente gerados usando outras ferramentas de BI. Após carregados, os dados são divididos em subconjuntos de DW, os Data Marts, que são usados para fornecer dados ao utilizadores (McKinsey & Company, 2011).

2.1.1. Métricas e Key Performance Indicators (KPI)

Num artigo do jornal Emerald, escrito por um grupo de engenheiros da universidade de Cambridge em 2005, afirmam que a questão da performance é muitas vezes discutida, mas raramente definida. Tratando a performance e as métricas de maneiras complemente distintas, "Literally it is the process of quantifying action, where measurement is the process of quantification and action leads to performance" (Neely, Gregory, & Platts, 2005).

Assumindo ainda as diferenças entre os seguintes termos:

Tabela 2.1 – Definições de Performance (Neely et al., 2005)

	Definição
Performance Measurement	"the process of quantifying the efficiency and effectiveness of action"
Performance Measure	"metric used to quantify the efficiency and/or effectiveness of an action"
Performance Measurement System	"Set of metrics used to quantify both the efficiency and effectiveness of actions."

Em suma, Andy Neely e o seu grupo reforçam que os PMS podem ser examinados em 3 níveis diferentes conforme mostra a figura 2.2:

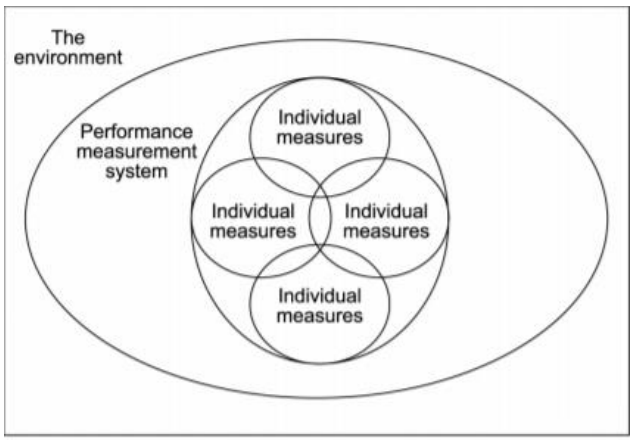


Figura 2.2 - A framework for performance measurement system design (Neely et al., 2005)

O primeiro nível passa pelas métricas individuais onde Neely reforça que uma métrica deve ser posicionada num contexto estratégico, uma vez que influenciam o que as pessoas fazem.

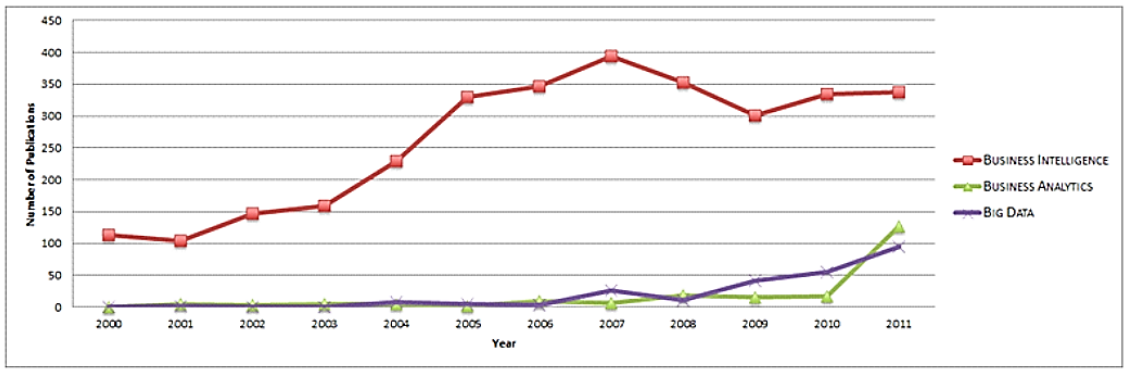
Por outro lado, uma métrica é definida por Barclay como (Barclay, 2015) "Business Metrics, sometimes referred to as Performance Metrics, measure the performance and activities of an organisation.". Rouge (Rouse, 2015) defende que "A business metric is a quantifiable measure use to track, monitor and assess the success or failure of various business processes." Esta definição é bastante elucidativa da função de uma métrica que passa por medir a performance e o desempenho do negócio. Porém uma métrica por si só não ajuda na tomada de decisão, isto é, tem de estar enquadrada num contexto de negócio para que faça sentido. A métrica é apenas um valor que não contém mais informação. Portanto, o contexto é o que permite uma métrica realmente ter impacto.

É neste âmbito que os KPI respondem a uma necessidade de melhorar a performance daquilo que se pretende medir. Barclay define (Barclay, 2015) "A Key Performance Indicator measures how effective the organisation is at achieving the business targets or strategy." Um KPI é um alvo a atingir, um objetivo a alcançar e é ele que dá sentido à métrica. Um dos aspetos fundamentais dos KPI prende-se com a sua correta definição. Roth alerta que (Roth, 2017) "Choosing the right KPI is crucial to make effective, data-driven decisions. If you choose the right KPI, it will help to concentrate the efforts of employees towards a meaningful goal, however, choose incorrectly and you could waste significant resources chasing after vanity metrics." Ou seja, a escolha e o enquadramento do KPI podem ser a diferença entre o sucesso e o fracasso.

Tendo em conta todos os aspetos propostos nos parágrafos anteriores, Neely no seu artigo (Neely et al., 2005) afirma que um dos problemas que existem quando se fala deste assunto é a diversidade tendo em conta as necessidades de cada leitor, pois estes tendem a focar-se em diferentes aspetos.

2.2. BIG DATA

Big Data é, hoje em dia, considerado um tópico bastante popular no mundo tecnológico, devido à necessidade das empresas de terem mais capacidades de armazenamento de informação. Durante décadas as companhias têm feito as suas decisões de negócio com base em dados transacionais armazenados em bases de dados relacionais, não desprezando esse tipo de dados, parte do proveito das vantagens competitivas de uma empresa pode estar em dados não estruturados como redes sociais, email e fotografias que se podem traduzir em informação útil (Dijcks, 2012). Contudo, tem sido um termo considerado ambíguo, controverso e com algumas definições contraditórias proveniente da falta de consistência nas definições divulgadas (Ward & Barker, 2013).



Keyword	All Years	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011
Business Intelligence	3,146	113	104	146	159	229	330	346	394	352	201	334	338
Business Analytics	213	0	5	43	4	5	2	9	6	19	16	17	126
Big Data	243	0	1	0	0	7	4	3	26	11	41	44	95
Total	3,602	113	110	149	163	241	336	358	426	382	358	356	560

Figura 2.3 – Tendência das publicações entre 2000 e 2011 (Chen & Storey, 2012)

O Big Data converteu-se num desafio em crescimento que as organizações enfrentam ao lidar com as grandes quantidades de dados e com o rápido crescimento das fontes de dados ou informação que apresentam complexidade analítica. Descrevendo assim uma nova geração de tecnologias e arquiteturas desenhadas para economicamente extrair valor de grandes volumes e variedades de dados enaltecendo a rapidez na recolha, descoberta e análise dos dados (Villars & Olofson, 2014).

A Oracle através de Jean-Pierre Dijcks, define Big Data como sendo um conjunto de vários tipos de dados, como dados tradicionais de uma empresa gerados pelos CRM's, ERP's e/ou armazenamentos de transações web, os dados sensoriais como os CDR e dados de sistemas (logs), e dados sociais provenientes das redes sociais como o Facebook ou Twitter (Dijcks, 2012).

Enquanto que a Oracle associa uma definição de Big Data à combinação de dados transacionais a dados não estruturados de modo a melhorar as decisões de negócio das empresas. O Gartner através de Laney, em 2001, usa uma definição referindo a existência de 3V's: Volume, Velocidade e Variedade que mais tarde, em 2012, juntamente com a IBM incluiu mais um "V", a Veracidade (Dijcks, 2012; Laney, 2001; Ward & Barker, 2013) .

A Mckinsey também definiu o conceito de forma bastante interessante, considerando Big Data um conjunto de dados cujo o tamanho é bastante superior a uma base de dados típica de ferramentas de software para capturar, armazenar, gerir e analisar os dados. Afirmando que não definem Big Data dando um valor em concreto de quantidades de tamanhos das Bases de Dados, pois esses mesmos valores podem mudar tendo em conta os sectores de atividade e porque ao longo do tempo esse mesmo valor iria aumentar (McKinsey & Company, 2011).

Assim sendo, o termo Big Data tem vindo a ser associado ao grande volume de dados, à rapidez com que os dados estão em mudança, à variedade de fontes que existem para extrair informação e à Veracidade dos dados formando assim os 4'V defendidos por Dumbill (Dumbill, 2012; Ribeiro, 2014). A adoção do Big Data é uma necessidade presente, (McKinsey & Company, 2011) "all sectors in the US economy had at least an average of 200 terabytes of stored data", esta estatística que remonta a 2009 cresceu exponencialmente até aos dias de hoje. Esta tecnologia está a revolucionar a forma como vemos o mundo, (Science, n.d.) "Big data will change the way we think about business, health, politics, education, and innovation in the years to come".

Tendo em conta o volume de dados e segundo um estudo da IDC, estima-se que até 2020 o tamanho do universo digital duplique a cada ano assim como o investimento das empresas em infraestruturas, como hardware e software, capazes de suportar a quantidade de informação gerada. Estimando-se um aumento de 40% no volume de dados no setor das telecomunicações (Gantz & Reinsel, 2012).

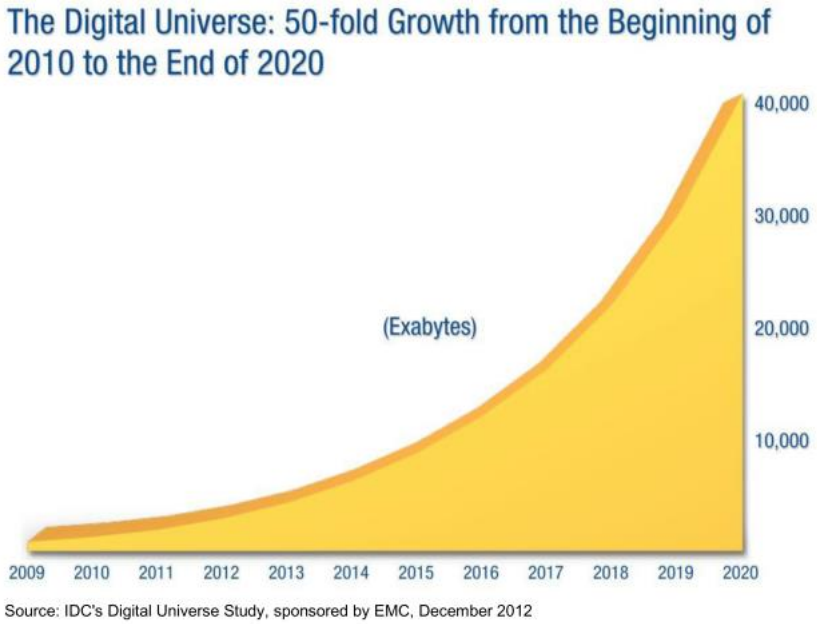


Figura 2.4 - Evolução do Universo Digital entre 2010 e 2020 pela IDC (Gantz & Reinsel, 2012)

As máquinas estão a gerar grandes quantidades de dados comparativamente com as fontes de dados tradicionais, "a single jet engine can generate 10TB of data in 30 minutes. With more than 25,000 airline flights per day, the daily volume of just this single data source runs into the Petabytes" (Dijcks,

2012). A McKinsey acrescenta que o volume de dados está a crescer cerca de 40% por ano e cresce 44 vezes mais entre 2009 e 2020 (McKinsey & Company, 2011).

Por o outro lado, Laney tem um pensamento derrotista em relação ao aumento do volume dos dados defendendo que com o aumento exponencial do volume de dados leva a um decréscimo proporcional do valor dos dados (Laney, 2001).

A Velocidade é outra característica do Big Data que se referente à possibilidade de transmissão de dados em tempo-real e a melhoramentos em termos de performances de plataformas, como por exemplo na área do e-commerce a resposta em termos de inventários, análises e seguimento de encomendas seria muito mais vantajoso (Laney, 2001). “Even at 140 characters per tweet, the high velocity (or frequency) of Twitter data ensures large volumes (over 8 TB per day)” (Dijcks, 2012).

Outra característica bastante importante é a variedade de dados que podem ser armazenados para posteriormente serem analisados. Enquanto que os dados estruturados mudam de maneira lenta, os dados não estruturados estão em constante mudança à medida que novos serviços são adicionados ou para novas campanhas de marketing em que são necessários recolher novos tipos de dados para tornar toda essa informação relevante (Dijcks, 2012).

Segundo um artigo da Faculdade do Rio de Janeiro, a Veracidade prende-se com a qualidade dos dados. Esta é essencial para os utilizadores interessados os usarem e reutilizarem da maneira mais apropriada, de modo a gerarem a sua própria informação (Ribeiro, 2014).

A Oracle entende que o quarto V é o Valor, pois no meio de tantos dados existe informação relevante escondida. O principais desafio é identificar o que poderá trazer valor e transformar e extrair esses dados para análise (Dijcks, 2012).

Tendo em conta estas características, através da Big Data é interessante estudar o caso da evolução do mundo das Telecomunicações, pois recentemente a maior parte dos dados gerados eram provenientes de chamadas efetuadas e recebidas e o tamanho das mesmas. Com o aparecimentos de smartphones e tablets novos dados foram adicionados com informação geográfica, mensagens de texto, internet e, até mesmo, emoções (Villars & Olofson, 2014).

2.2.1. Hadoop

Segundo o site Apache Hadoop, “The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures” (Hadoop, n.d.).

A McKinsey também vai ao encontro do site do Apache, definindo o Hadoop como “An open source (free) software framework for processing huge datasets on certain kinds of problems on a distributed system. Its development was inspired by Google’s MapReduce and Google File System. It was originally developed at Yahoo! and is now managed as a project of the Apache Software Foundation” (McKinsey & Company, 2011).

Hadoop é a abreviatura de “Highly Archived Distributed Object Oriented Programming” criado em 2005 por Goug Cutting e Mike Cafarella para suportar um projeto de motores de pesquisa distribuídos. É uma tecnologia estruturada em Java que ajuda a armazenar, aceder e a obter grandes recursos a partir de Big data de maneira distribuída a um custo menor, alto grau de tolerância a falhas e alta escalabilidade (Saraladevi et al., 2015).

O Hadoop e o seu ecossistema são bastante conhecidos pelo MapReduce e o seu filesystem distribuído, o HDFS (Hadoop Distributed FileSystem). O MapReduce consiste num “distributed data processing model and execution environment that runs on large clusters of commodity machines (…) MapReduce works by breaking the processing into two phases: the map phase and the reduce phase. Each phase has key-value pairs as input and output, the types of which may be chosen by the programmer. The programmer also specifies two functions: the map function and the reduce function” e o HDFS pode ser definido como “ A distributed filesystem that runs on large clusters of commodity machines” (White, 2010).

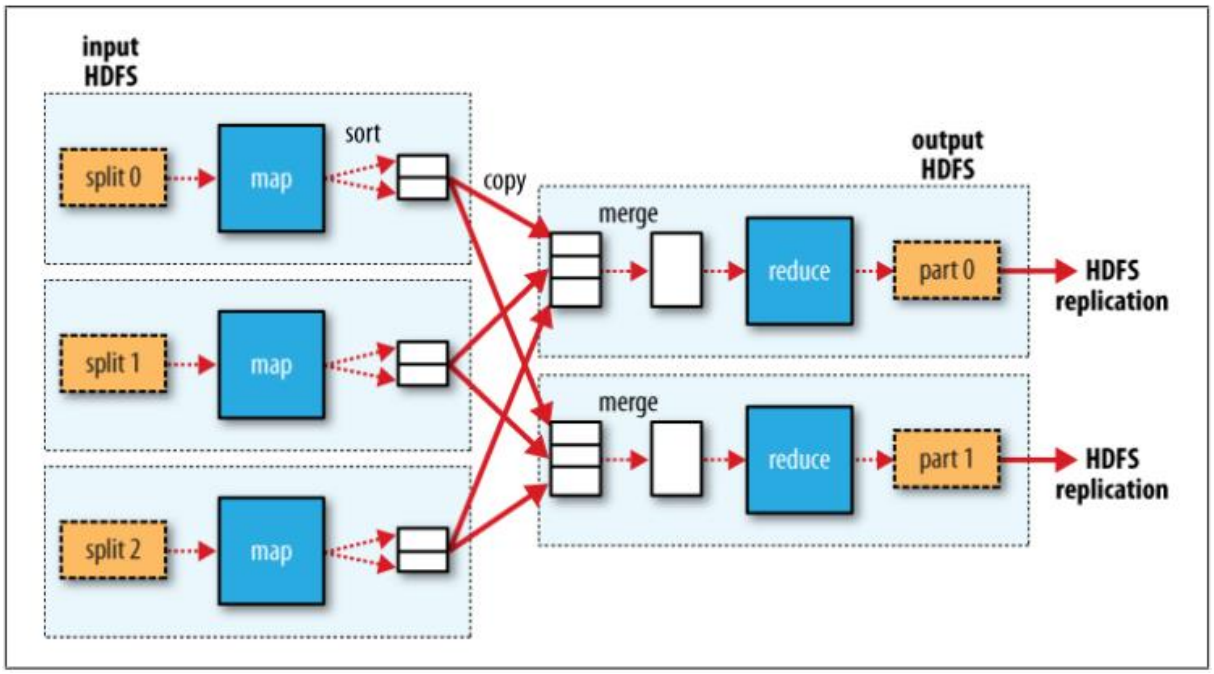
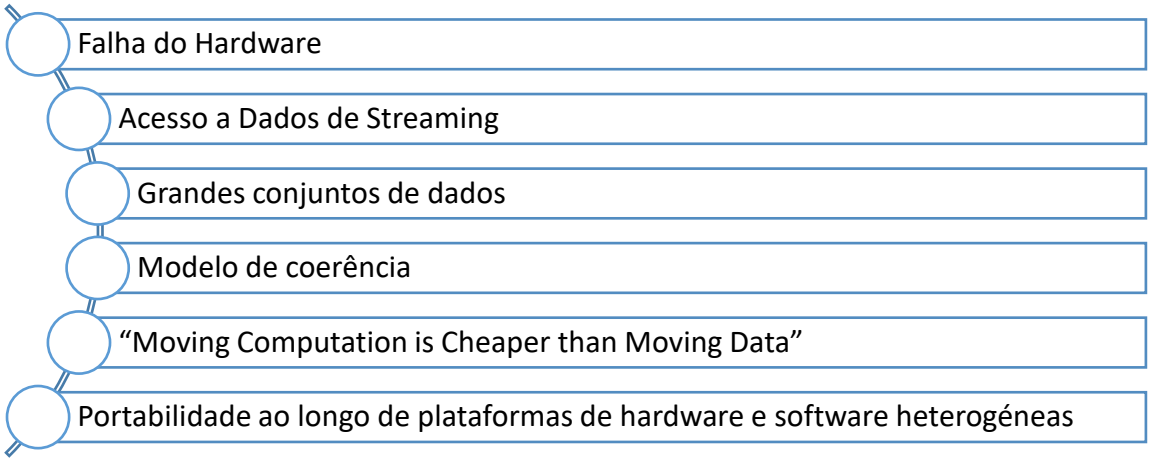


Figura 2.5 – Fluxo de MapReduce (White, 2010)

Segundo o site oficial da Apache Hadoop, o HDFS é usado com base em hipóteses e objetivos (Hadoop, n.d.):



Para além do MapReduce e do HDFS existem outros projetos como Pig, Hive, Sqoop e Spark.

O Pig é uma linguagem de fluxos de dados de alto nível e com execuções computacionais em paralelo (Hadoop, n.d.). White completa ainda, que o Pig serve para explorar grandes conjuntos de dado, pois corre no HDFS e em clusters de MapReduce (White, 2010).

O Hive é considerado um software de Data Warehouse que facilita na leitura, escrita e gestão de grandes conjuntos de dados armazenados através do SQL (Hive, 2018). Permite aos utilizadores fazerem a conversão de scripts em MapReduce em queries (Thusoo et al., 2010).

Por outro lado, o Sqoop é uma ferramenta de importação de base de dados relacionais para o Hadoop (HDFS) (Sqoop, 2018).

Por último, o Spark é um simples modelo de programação que suporta um conjunto de aplicações sendo as mais importantes o ETL e Machine Learning (Hadoop, n.d.). Conseguindo correr programas 100 vezes mais rápido que o MapReduce, tendo sido considerado o motor mais rápido para processamento de dados de grande escala (Spark, 2018).

O Apache Hadoop é considerado atualmente uma das melhores ferramentas para processamento de alta demanda de dados. Contudo, existem algumas desvantagens como o facto de estar constante evolução e algumas de suas funcionalidades estarem numa fase precoce (Goldman, Kon, Junior, Polato, & Pereira, 2012).

2.2.1.1. Hadoop Distributed File System

O HDFS apesar de ter muitas semelhanças com algumas ferramentas de armazenamento de ficheiros, as diferenças entre elas ainda são consideravelmente significantes. Entrando em mais detalhe e como referido a cima pelo Apache Hadoop, as diferenças das restantes ferramentas para o HDFS baseia-se em (Hadoop, n.d.):

Tabela 2.2 – Características do HDFS

Características	Descrição
Falha do Hardware	Uma vez que o HDFS é constituído por dezenas de servidores e cada um deles armazena uma parte do sistema, existe sempre a possibilidade de um desses servidores falhar. Tornando a deteção rápida dessas falhas e a recuperação automática um dos objetivos da arquitetura.
Acesso a Dados de Streaming	É desenhado para um maior processamento em carga do que interações com os utilizadores. Fazendo com que o acesso aos dados seja efetuado com maior rendimento.
Grandes conjuntos de dados	Está preparado para suportar grandes quantidades de dados que tipicamente variam entre os gigabites e os terabites. Suporta milhões de ficheiros apenas numa instância.
Modelo de coerência	Segue a premissa de "write-once-read-many". Um ficheiro que já foi criado e escrito já não pode ser modificado, podendo apenas ser eliminado. Simplificando a coerência dos dados.
"Moving Computation is Cheaper than Moving Data"	Quaisquer computações requeridas pelas aplicações são mais eficientes perto dos dados que operam. Minimizando o congestionamento da rede e aumenta a performance do sistema. Sendo melhor migrar os dados para onde a computação está a ser executada, em vez de mover os dados para onde aplicação está a correr.
Portabilidade ao longo de plataformas de hardware e software heterogéneas	Está desenhado para ser facilmente transportado entre plataformas.

A sua arquitetura é composta por dois tipos de nós, O Namenode que é considerado o mestre e um conjunto de Datanodes que são considerados os escravos ou trabalhadores (White, 2010)

O Namenode gere os namespaces do sistema e regula o acesso dos clientes aos ficheiros através da criação de blocos e da replicação após as instruções dos Datanodes. Enquanto que os Datanodes gerem o armazenamento dos nós sendo responsáveis por servirem os pedidos de leitura e escrita por parte dos clientes. Dentro da arquitetura, sempre que um ficheiro entra no Namenode passa por um processo de divisão em um ou mais blocos acabando armazenado nos Datanodes (Hadoop, n.d.)

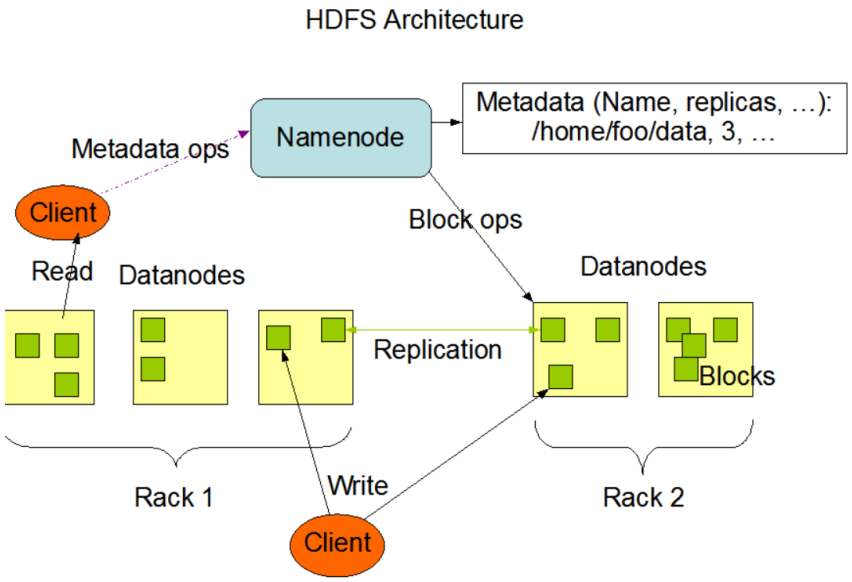


Figura 2.6 – Arquitetura HDFS (Hadoop, n.d.)

Visto que um dos objetivos do HDFS é armazenar dados tendo em conta a presença de fracasso, os tipos de falhas mais comuns são (Hadoop, n.d.):

Falhas no disco e re-replicação

Cada DataNode envia mensagens, chamadas de Heartbeat, para o Namenode de forma periódica. A partição da rede pode fazer com que um conjunto de DataNodes perca a conexão com o Namenode. Assim que o Namenode deteta a ausência dessas mensagens marca o Datenode como "morto" e não envia mais nenhum pedido para os mesmos. Assim que os DataNodes são marcados, os dados que lá registados não estarão disponíveis no HDFS. Apesar de o Namenode estar constantemente a verificar quais dos blocos precisam de ser replicados, a necessidade de re-replicação existe devido a:

- O Datanode pode ficar indisponível
- A replicação pode ficar num estado corrompido
- O disco ou o Datanode podem falhar
- O fator de replicação pode aumentar

Rebalanceamento do Cluster

A arquitetura do HDFS é compatível com os dados a rebalancear entre esquemas podendo mover automaticamente os dados de um Datanode para outro se o espaço livre atingir determinado patamar. No caso de um ficheiro em particular ter muita procura o schema pode dinamicamente criar réplicas adicionais e rebalancear outros dados no cluster

Integração dos Dados

Como verificado a cima os dados enviados pelo Datanode podem estar corrompidos. Sendo causado por possíveis falhas no dispositivo de armazenamento, falhas na rede ou problemas no software. Sempre que um ficheiro é criado no HDFS é gerado um código, checksum, para cada bloco onde o novo ficheiro será armazenado. Cade vez que o conteúdo desse ficheiro é requisitado o HDFS verifica se os dados recebidos de determinado ficheiro de cada Datanode são iguais ao checksum inicialmente gerado. Caso não aconteça o cliente pode optar por ir buscar os dados a outro datanode que contém a replica do bloco requisitado.

Falha no disco da Metadata

O HDFS tem como estruturas centrais o FSImage e o EditLog. Caso um destes ficheiros esteja corrompido, HDFS poderá ficar inoperacional. Para evitar este problema é possível configurar o Namenode de modo a suportar múltiplas cópias e a sincronização desses ficheiros em casos de mudanças. Porém, as atualizações às múltiplas cópias podem degradar as transações dos namespaces que um namenode pode suportar, pois a metadata não é intensiva.

A máquina do Namenode é o único ponto de falha para um cluster, pois se esta falha é necessária intervenção manual.

Snapshots

Os Snapshots suportam o armazenamento de uma cópia de dados num determinado período do tempo. Uma das utilizações dos Snapshots pode ser para reverter uma instância do HDFS corrompida para um ponto no tempo conhecido anteriormente.

Contudo, White afirma que é necessário analisar os casos em que o HDFS não funciona como esperado. Considerando os seguintes cenários (White, 2010):

Tabela 2.3 – Cenários de Inoperabilidade do HDFS

Cenários	**Descrição**
Acesso aos Dados de Baixa Latência	Aplicações que requerem baixa latência no acesso aos dados não vão funcionar corretamente no HDFS. O HDFS é otimizado para

	entregar um grande conjunto de dados e isso pode ser atingido às custas da latência.
Existência de Muitos Ficheiros de Tamanho Pequeno	Visto que o Namenode guarda a Metadada do repositório de dados em memória, o limite para o número de ficheiros no sistema é gerido pela quantidade de memória do namenode. Como regra geral, cada arquivo, diretório e bloco leva cerca de 150 bytes.
Múltiplas Escritas e Modificação Arbitraria de Ficheiros	Os ficheiros no HDFS podem ser escritos por um único escritor. As escritas são sempre feitas no final de cada ficheiro. Não existe suporte para vários escritores ou para modificações arbitrárias no ficheiro.

No entanto, existem alguns métodos que não são ainda suportados como: o Rebalanciamento, restarts automáticos, failouvers e os Snapshots não são suportados, mas Apache Hadoop garante que o serão em versões futuras.

2.2.1.2. Hive

Como referido anteriormente, O Hive é considerado um software de Data Wharehouse que facilita na leitura, escrita e gestão de grandes conjuntos de dados armazenados através do SQL (Hive, 2018). Estruturando os seus dados em conceitos bastante conhecidos de Base de Dados como tabelas, colunas, linhas e partições. Suporta também a maior parte dos tipos de dados primitivos como integers, floats, doubles e strings bem como alguns tipos complexos como maps, lists e structs (Thusoo et al., 2010).

O Hive implementa uma linguagem do tipo SQL chamada HiveQL (Floratou, Minhas, & Ozcan, 2014).

Em termos de arquitetura, como ilustrado na figura 2.7, é composto pelos seguintes componentes:

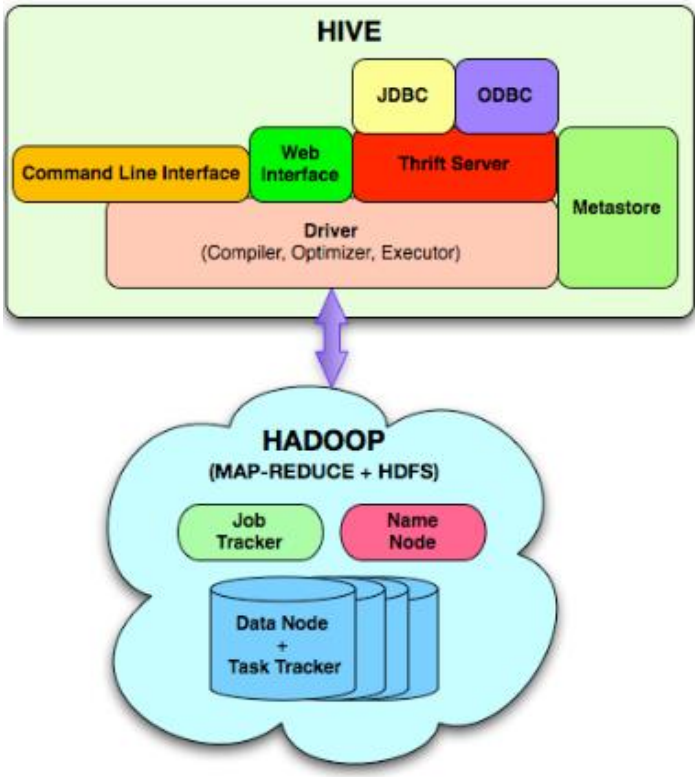


Figura 2.7 – Arquitetura Hive (Thusoo et al., 2010)

Segundo Thusoo os componentes que são considerados os grandes componentes do Hive são (Thusoo et al., 2010):

- **Metastore**
  - Componente que armazena um sistema catálogo que contém toda a metadata das tabelas existentes no sistema
- **Driver**
  - Gere o ciclo de vida dos códigos de HiveQL durante a sua compilação, otimização e execução.
- **Query Compiler**
  - Invoca o Driver uma vez que recebe o código. Traduz o código num plano.
- **Execution Engine**
  - Serve para executar as tarefas produzidas pelo compilador pela ordem de dependências correta.
- **HiveServer**
  - Componente que permite integrar o Hive com outras aplicações.
- Command Line Interface (CLI), web UI e JDBC/ODBC driver.
- Extensibility Interfaces

Quando um código de HiveQL é executado passa por um processo de conversão, compilação e otimização para produzir um plano de execução.

Na forma de armazenar dados, o Hive armazena os seus dados em tabelas associadas a diretorias do HDFS. Porém, as tabelas criadas em Hive têm algumas particularidades, pois uma tabela em Hive pode ser do tipo Internal ou External. As Tabelas Internal são as que utilizam o comando base "CREATE TABLE (…)" onde a execução de um comando drop nesse tipo de tabelas elimina não só a tabela como os dados associados no HDFS. As tabelas do tipo External utilizam como comando base "CREATE EXTERNAL TABLE (…)" onde a eliminação dessa tabela não leva à eliminação dos ficheiros associados. Outro tipo de armazenamento permitido no Hive são as partições, onde os dados são armazenados em subdiretorias dentro da diretoria da tabela. Por fim, podemos armazenar os dados em buckets que se caracterizam por ser um ficheiro dentro da partição ou diretoria da tabela dependendo se a tabela está particionada ou não (Thusoo et al., 2010).

Em termos de linguagem SQL, o Hive suporta select, joins, agregações, unions e subqueries. Permite ainda queries de inserts para que os utilizadores insiram carreguem os dados provenientes de diferentes fontes para as tabelas correspondentes. Contudo, existe a limitação de não se conseguir fazer updates ou eliminação de registos específicos nas tabelas (Thusoo et al., 2009).

No que toca à compressão das tabelas, utilizam um tipo de organização colunar que tem benefícios significantes na análise de queries. O Hive detém o seu próprio formato, Hive Optimized Row Columnar (ORC) (Floratou et al., 2014).

De acordo com o site do Apache ORC, os formatos ORC permitem um armazenamento eficiente através da codificação dos dados e criação de um índice interno, de modo a que o utilizador cada vez que fizer uma pesquisa sobre a tabela, o Hive descomprime e processa apenas os valores que são requeridos pelo código usado (apache, n.d.).

A ilustração abaixo, retirada de um artigo da Hortonworks, permite comparar os tipos de compressão existentes.

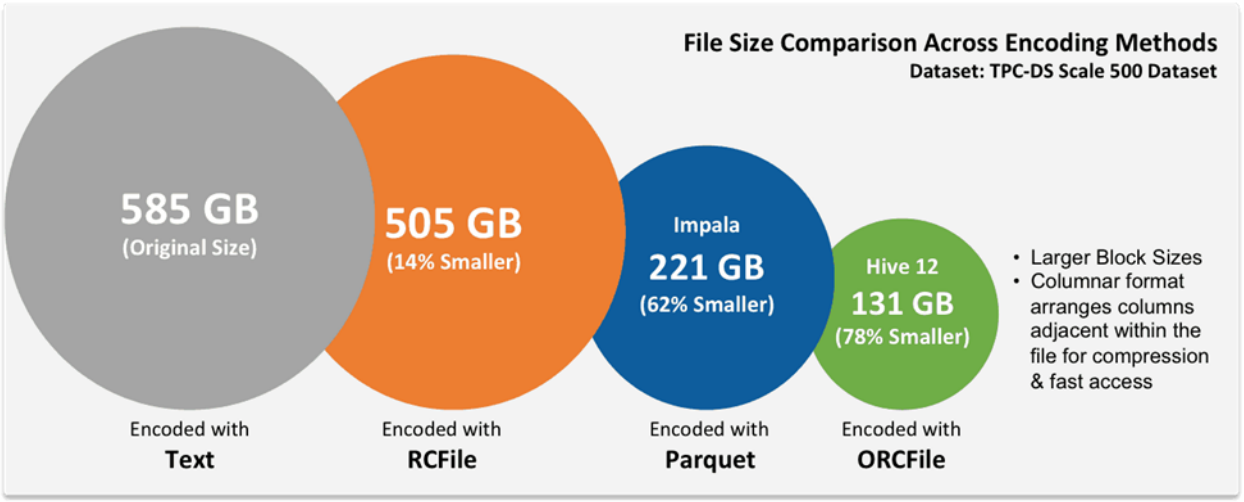


Figura 2.8 – Comparação dos vários tipos de formato (Hortonworks, n.d.)

2.2.1.3. Impala

O Impala é considerado uma ferramenta open-source de análise de Base de dados. Surgiu da necessidade de se aumentar a performance das queries em Hadoop. Possibilita serem feitas análises sobre dados armazenados no HDFS ou em HBase e operações em tempo real que mantém a experiência do utilizador familiar. De modo a manter a experiencia familiar, utiliza a mesma metadata, sintaxe que o HiveQL, drivers e interface permitindo aos utilizadores do Hive usar a ferramenta (Impala, n.d.).

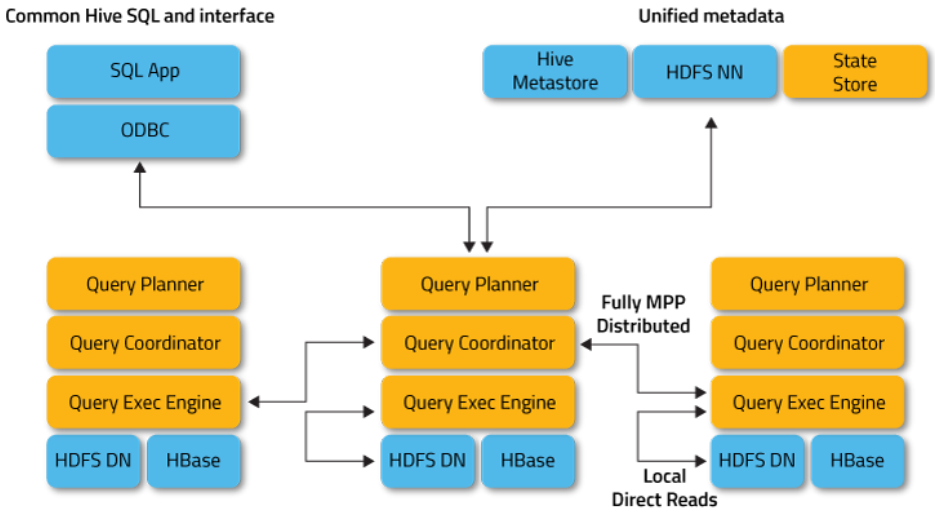


Figura 2.9 - Arquitetura Impala (Impala, n.d.)

A arquitetura representada na figura acima permite evitar a latência e contornar o MapReduce, de modo a aceder diretamente aos dados através de um mecanismo bastante semelhante aos que existem nos RDBMS. Permitindo assim obter melhor performance que o Hive dependendo do tipo de querie e configuração (Impala, n.d.).

Segundo a artigo redigido por uma equipa da Cloudera (Kornacker et al., n.d.), o Impala é um mecanismo massivo de execução paralela de queries. Explicando ainda, que o processo de implementação do impala é composto por três serviços:

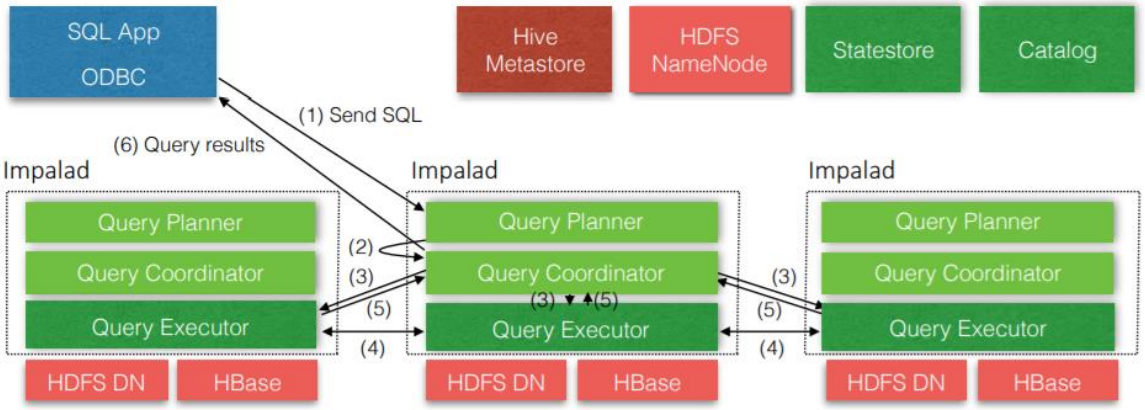


Figura 2.10 – Processo de Execução de Queries em Impala (Kornacker et al., n.d.)

- Impala Daemon (impalad) – Responsável por aceitar as queries, orquestrar as execuções ao longo do cluster e executar fragmentos das queries. Os Daemons ajudam na propriedade relacionada com a tolerância à falha e com o balanceamento dos carregamentos. Por defeito, existe um em cada máquina no cluster, o que permite tirar proveitos da localização dos dados e a ler blocos do filesystem sem usar a rede.
- Statestore daemon (statestored) – Serviço de metadata
- Catalog daemon (catalogd) – Serve como repositório de acesso à metadata. Através do catalogd, os Daemons podem executar comandos de DDL que estão refletidos num catálogo externo, como por exemplo, metadata do Hive.

O Impala é capaz de suportar formatos de ficheiros do tipo Avro, RC, Sequence, plain text, and Parquet Podendo ser combinados com diferentes tipos de algoritmos de compressão como snappy, gzip, bz2.

De acordo com a opinião de Kornacker e da sua equipa (Kornacker et al., n.d.), na maioria dos casos o tipo de formato recomendado é o Parquet. Um tipo de ficheiro de formato colunar de alta compressão e eficiência sendo compatível com Hive e MapReduce capaz de otimizar grandes blocos de dados.

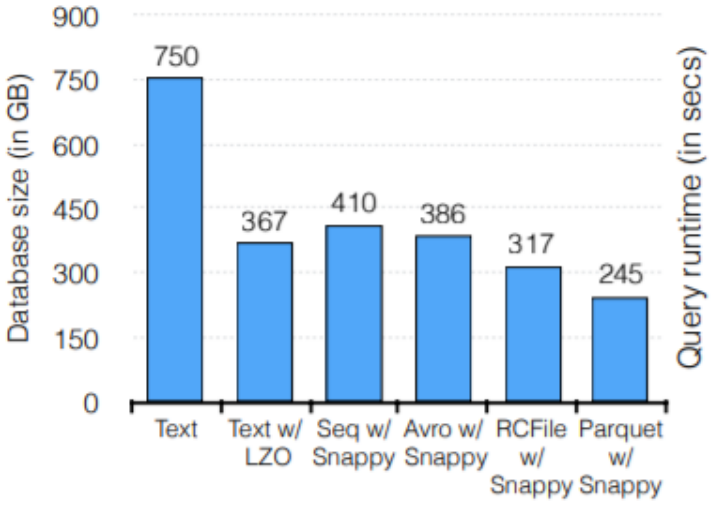


Figura 2.11 – Comparação dos Rácios de Compressão de tipos de formatos e compressões populares (Kornacker et al., n.d.)

Segundo o gráfico 2.11, o autor indica que os ficheiros de formato Parquet com compressão do tipo snappy atingem os melhores níveis de compressão entre os restantes tipos. Concluindo que este tipo consegue obter uma performance em cerca de cinco vezes melhor que os restantes.

Para finalizar, segundo o Apache Impala, existem vantagens no uso desta ferramenta de análise (Impala, n.d.):

- Com o processamento local em Datanode, os problemas da rede são evitados;
- Pode ser utilizada uma única, aberta e unificada metadata;
- A conversão dispendiosa de dados é desnecessária;
- Todos os dados são imediatamente consultáveis, sem atrasos para o ETL;
- Todo o hardware é utilizado para consultas do Impala, bem como para o MapReduce;
- Apenas uma única máquina é necessária para escalar;

2.3. Comparação de Arquiteturas

As maiores tecnologias que sustentam o BI são os Sistemas de Gestão de Bases de Dados, Data Warehousing, processos de ETL, OLAP e BPM (Chen & Storey, 2012).

Numa estrutura típica de BI os dados sobre os quais as tarefas serão efetuadas são carregados num repositório chamado Data Warehouse (DW) que é gerido por um ou mais servidores DW. A escolha mais frequente para armazenar e analisar estes dados é através de um modelo relacional de sistema de gestão de base de dados (RDBMS) (Chaudhuri, Dayal, & Narasayya, 2011).

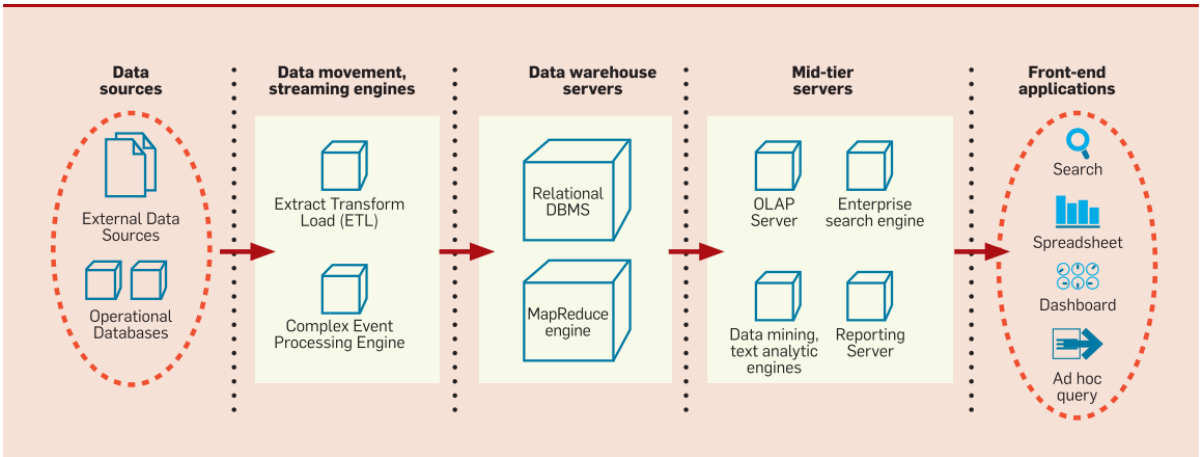


Figura 2.12 - Arquitetura Tradicional de BI (Chaudhuri et al., 2011)

Data Warehousing é um conjunto de tecnologias capaz de ajudar nas tomadas de decisão com o objetivo de permitir ao utilizador tomar melhores e mais rápidas decisões (Webb, 2015).

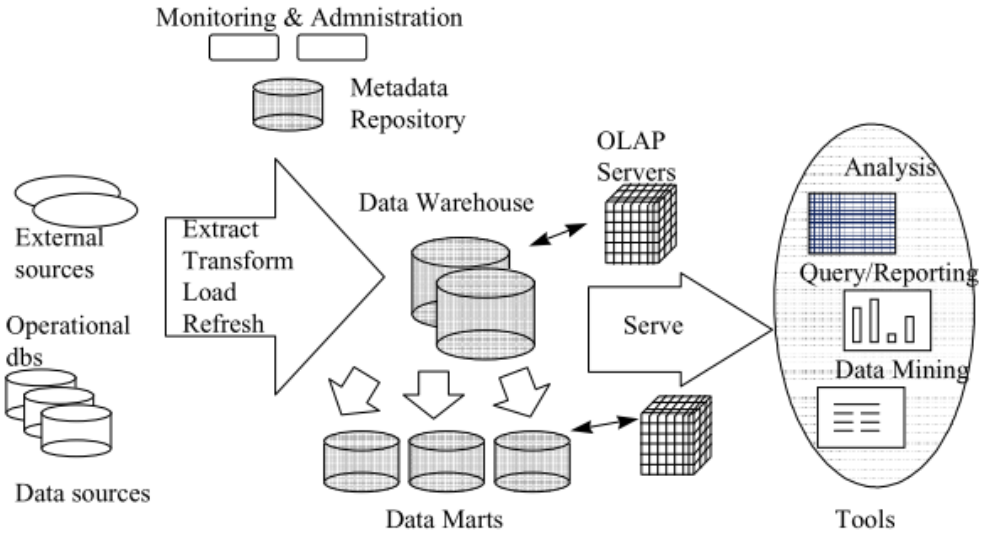


Figura 2.13 – Processo de Data Warehousing (Webb, 2015)

Tipicamente um DW é mantido separadamente da base de dados operacionais das empresas. O DW suporta o OLAP com os requerimentos funcionais e de performance e o OLTP que é suportado por base de dados operacionais. Tendo por objetivo ajudar na tomada de decisão fazendo com que os dados históricos, sumarizados e consolidados sejam mais importantes do que os registos individuais. Os fluxos de carregamento são caracterizados por queries intensivas e complexas que podem aceder a milhões de registos e efetuar muitos pesquisas, joins e agregações. Sendo mais importante os tempos de resposta do que os tempos de transações. De modo a facilitar essas análises e visualização dos dados, o DW é tipicamente modelado de forma multidimensional (Webb, 2015).

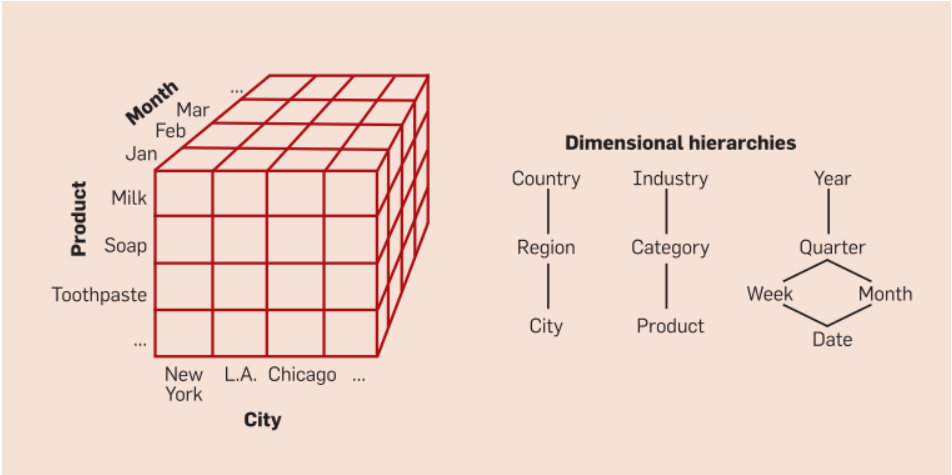


Figura 2.14 - Modelos Multidimensionais (Chaudhuri et al., 2011)

São considerados complementos ao DW, que fornecem funcionalidades especializadas para diferentes cenários BI, os seguintes elementos:

OLAP

Chaudhuri (Chaudhuri et al., 2011) afirma que o OLAP expõe eficientemente a visão multidimensional dos dados para as aplicações ou utilizadores. Permitindo a execução de operações comuns do BI como filtros, agregações, drill-downs e pivoting.

É caracterizado por operações de (Webb, 2015):

- Rollup - quando aumentamos o nível de agregação
- Drilldown – Quando diminuímos o nível de agregação e consequentemente aumentamos o nível de detalhe
- Slice and Dice – Quando se executam operações de seleção e projeção
- Pivot – Reorientação da visão multidimensional dos dados

OLTP

É caracterizado por aplicações que automaticamente executam tarefas de processamento de dados administrativos que são considerados o dia-a-dia das operações de uma organização. Essas tarefas são estruturadas e repetitivas e consistem em transações consistentes, isoladas e atómicas. Estes tipos de dados são armazenados em bases de dados operacionais onde a consistência e a recuperabilidade é crítica (Webb, 2015).

Á medida que mais dados aparecem na forma digital, aumenta a necessidade de se desenharem novas arquiteturas de plataformas baratas que podem suportar volumes de dados muitos maiores que os tradicionais RDBMS. Este paradigma é conhecido como o desafio do Big Data (Chaudhuri et al., 2011).

Dado este facto, mecanismos baseados em MapReduce que inicialmente foram criados para analisar documentos e fazer pesquisas Web estão agora a ser utilizados por analistas empresariais. Estando essas novas ferramentas a ser estruturadas para suportar queries de SQL (Chaudhuri et al., 2011).

Segundo Dijcks, quando Big Data é extraído e analisado em conjunto com os mecanismos tradicionais, as empresas podem desenvolver uma compreensão mais completa e perspicaz do seu negócio. Levando posteriormente a um aumento da produtividade, aumento da competitividade e evolução em termos de inovação (Dijcks, 2012).

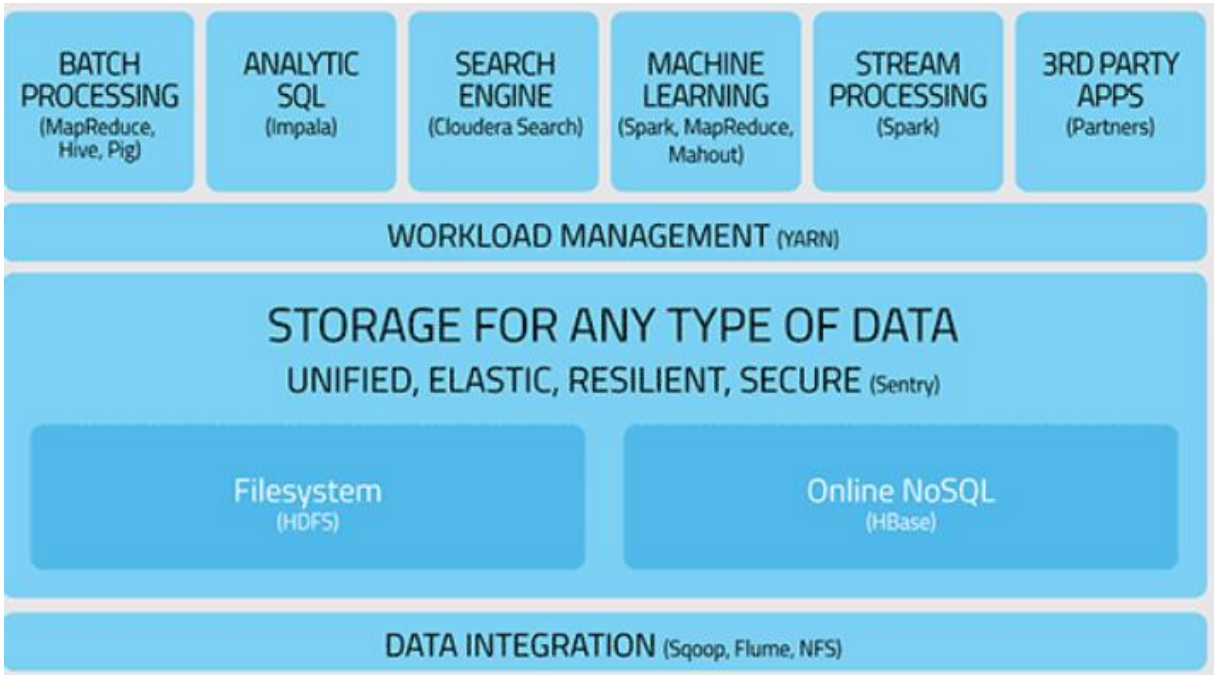


Figura 2.15 - Arquitetura Hadoop (Cloudera, n.d.)

Dijcks acrescenta que uma arquitetura de Big Data tem requisitos únicos diferentes dos tradicionais, uma vez que o objetivo é integrar facilmente os dados, de modo a levar a análises profundas em conjuntos de dados combinados (Dijcks, 2012).

Chaudhuri (Chaudhuri et al., 2011) afirma que as plataformas baseadas em repositório de ficheiros distribuídos e usando o MapReduce em tempo real têm sido um sucesso. Tendo sido implementadas em clusters com mais nós que os tradicionais DBMS. Ao contrário dos DBMS paralelos, onde os dados são primeiro carregados nas tabelas num schema predefinido antes de ser executadas as análises. O MapReduce pode ser executado diretamente nos ficheiros sendo capaz de suportar problemas como particionamento e falha dos nós, gestão dos fluxos ao longo dos nós e a heterogeneidade dos nós.

Outro fator que torna esta plataforma atrativa é a habilidade de suportar análises em dados não estruturados, imagens e dados sensoriais. Recentemente, estes mecanismos foram estendidos para suportar empresas como é o caso da Cloudera.

Contudo, esta abordagem ainda se encontra numa etapa prematura comparada com os sistemas tradicionais RDBMS, embora a sua exploração esteja a crescer rapidamente pelo facto da existência de ferramentas open source, como é o caso do ecossistema hadoop.

Dijcks explica em termos de organização, que nos clássicos DW a organização dos dados é caracterizada por "Data Integration", pois existe um grande volume de dados e a tendência de organiza-los nos seus destinos poupando tempo e dinheiro com o facto de não se mover ao longo da estrutura grandes quantidades de dados. Por outro lado, o Hadoop (como explicado nos capítulos anteriores) é uma tecnologia que permite a organização e processamento desses grandes volumes de dados mantendo os dados originais guardados no cluster (Dijcks, 2012).

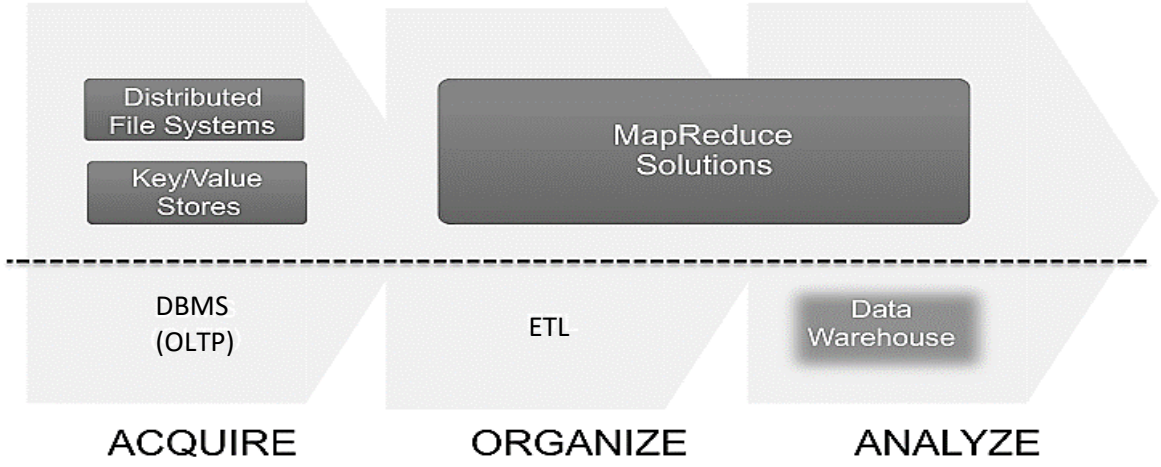


Figura 2.16 - Comparação entre Arquiteturas DW e Hadoop (Dijcks, 2012)

2.3.1. Relação entre Queries Hadoop e SQL Oracle

Tendo em conta que os softwares a serem usados têm como base a construção de queries em SQL Oracle na maioria das suas funcionalidades. O presente capítulo sugere de que modo é que a linguagem teria de ser adaptada e esta nova realidade.

Para isso, comparar a linguem SQL Oracle com Hive e Impala é imperativo visto que, como foi referido anteriormente, são dois serviços de construção de queries em Hadoop.

Segundo uma definição da McKinsey Global Institute, o SQL é "Originally an acronym for structured query language, SQL is a computer language designed for managing data in relational databases. This technique includes the ability to insert, query, update, and delete data, as well as manage data schema (database structures) and control access to data in the database" (McKinsey & Company, 2011).

As empresas estão a usar o Hadoop como repositório central para todos os dados que venham de diferentes fontes quer sejam estruturados ou não estruturados com o objetivo de gerir e correr análises profundas para extrair conhecimento dos dados (Floratou et al., 2014).

SQL Oracle é um conjunto de linguagens declarativas que permite uma interface num sistema relacional de gestão de base de dados, mais conhecidos por RDBMS. Sendo considerado uma linguagem universal estandardizada segundo afirma Sethy e a sua equipa. É demonstrado na figura 2.17 todos os passos do processamento de uma query em SQL Oracle. Encontra-se dividido em quatro estados: conversão, otimização, geração de linhas e execução de comandos (Sethy, Dash, & Panda, 2018).

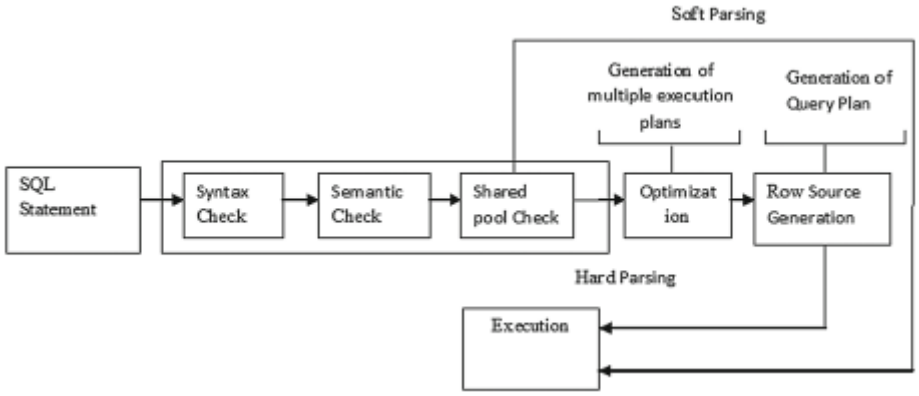


Figura 2.17 - Estados do Processamento de Queries em SQL Oracle (Sethy et al., 2018)

Ao longo dos últimos tempos, várias otimizações, estruturações e técnicas de processamento foram feitas para executar análises complexas em grandes quantidades de dados. De modo a suportar de forma eficiente operações como Filtros, Joins e agregações para dar resposta a tomadas de decisões alguns mecanismos de armazenamento dados devem ser tidos em conta (Chaudhuri et al., 2011):

Estrutura de Índices:

Os utilizadores acedem aos dados de forma associativa com base nos valores de uma determinada coluna. Facilita-se desta forma as pesquisas quando num comando se usa uma condição de filtros. O uso destas operações pode reduzir significativamente ou até mesmo em alguns casos eliminar o acesso as tabelas base. Sendo muito eficientes para domínios de cardinalidade baixa.

Particionamento:

O particionamento dos dados pode ser usado para melhorar a performance e a capacidade de gestão. Permitindo que tabelas e índices sejam divididos em unidade mais pequenas mais controladas. Ajudando na manutenção de operação de base de dados como carregamento e backups que pode ser executado sobre as partições em vez de tabelas ou índices.

Como foi referido no capítulo anterior por Chaudhuri (Chaudhuri et al., 2011), devido ao facto dos DW sentirem a necessidade de serem capazes de executar queries complexas em SQL o mais eficientemente possível sobre um grande volume de dados, o autor afirma que a chaves para a solução

deste problema passa pela otimização das queries. A otimização das queries é caracterizada pela compilação das queries complexas num plano de execução.

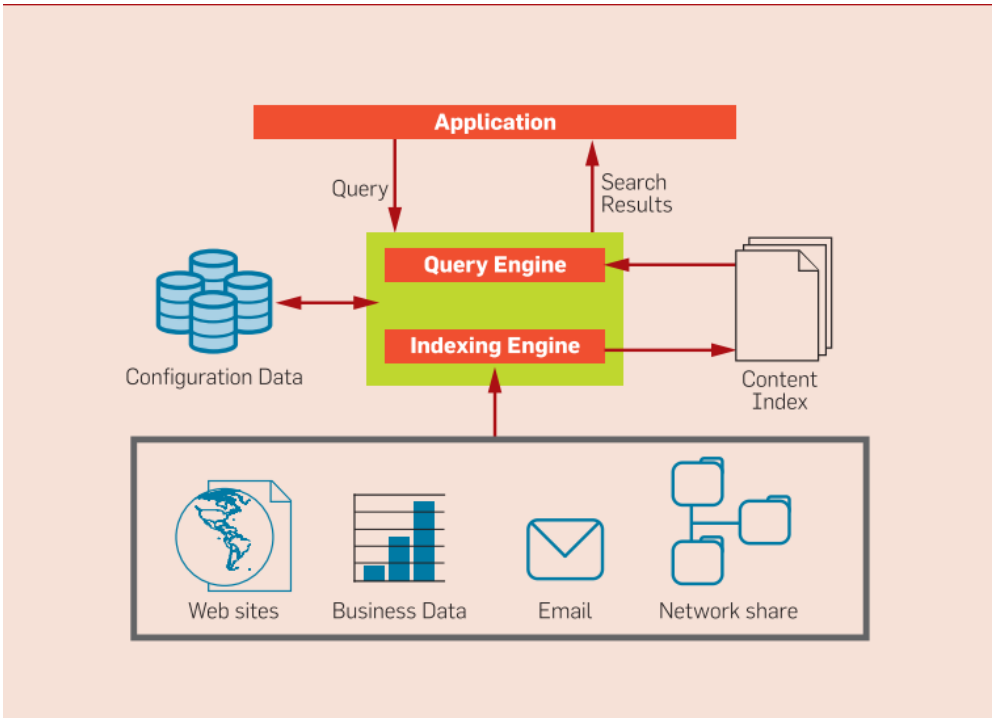


Figura 2.18 - Arquitetura de Pesquisa das Empresas (Chaudhuri et al., 2011)

White, no seu livro (White, 2010), dedica um capítulo apenas a esta temática começando por afirmar que as bases de dados tradicionais seguem um design chamado “Schema on Write”, pois se os dados que estiverem a ser carregados não estiverem em conformidade com o Schema são rejeitados. No entanto, a performance das queries é mais rápida, uma vez que a base de dados pode conter índices e tipos de compressão. Porém, o Hive segue o design “Schema on Read”, pois cada vez que existe uma inserção de dados estes não são verificados. Este fenómeno torna o Hive mais rápido, dado que não têm de ler, converter e serializar os dados no disco.

Tendo em conta a arquitetura já demonstrada no capítulo 2.2.1.2 e num artigo baseado na migração feita de aplicações SQL para Hive escrito por Wang e a sua equipa do Institute of Computing Technology (Wang, Xu, Liu, Chen, & Hu, 2015), os principais desafios a considerar são:

- O Hive não consegue suportar totalmente a sintaxe do SQL. Um exemplo são os comandos de Update ou Delete. De modo a contornar esse desafio e modificar ficheiros existentes, a solução passaria por reescrever o ficheiro na integra
- Apesar de algumas queries poderem ser diretamente aceites no Hive, a sua performance pode-se tornar lenta por devido à diferença dos modelos de funcionamento RDBMS e do MapReduce

Completando o primeiro ponto descrito por Wang, White explica que a falha de Updates, transações e índices no Hive deve-se ao facto de este operar sobre o HDFS usando o MapReduce onde a pesquisa

de todos os dados é considerada norma e a atualização das tabelas é atingida através da transformação de dados para uma nova tabela (White, 2010).

Na sequência de um artigo criado com base na infraestrutura do Facebook (Thusoo et al., 2010), Thusoo afirma que a linguagem do Hive (HiveQL) contém algumas características do SQL tradicional como "sub- queries, various types of joins – inner, left outer, right outer and outer joins, cartesian products, group bys and aggregations, union all, create table as select and many useful functions on primitive and complex types" fazendo com que qualquer individuo familiarizado com o SQL seja capaz de trabalhar com o Hive.

Por um lado, Thusoo e a equipa encontraram limitações no modo como o carregamento de dados através de comandos de INSERT são feitos. Referindo, também, a falta de comandos como UPDATE ou DELETE (Thusoo et al., 2010).

Por outro lado, á semelhança do que acontece com o SQL Tradicional, O Hive também contém um compilador que gera os planos de execução através dos seguintes passos:

- Conversão – Criação de uma árvore de sintaxe
- Verificação Semântica – O compilador vai buscar a informação das tabelas ao Metastore e usa essa informação para contruir um plano lógico. Verifica também a compatibilidade das expressões e informa no caso da existência de algum erro.
- Otimização – consiste na cadeia de transformações

Em suma, a seguinte tabela representada na figura 2.19 sumariza tudo o que foi descrito ao longo do presente capítulo.

	RDBMS	Hadoop
Data sources	Structured data with known schemas	Unstructured and structured
Data type	Records, long fields, objects, XML	Files
Data Updates	Updates allowed	Only inserts and deletes
Language	SQL & XQuery	Pig (Pig Latin), Hive (HiveQL), Jaql
Processing type	Quick response, random access	Batch processing
Data integrity	Data loss is not acceptable	Data loss can happen sometimes
Security	Security and auditing	Partial
Compress	Sophisticated data compression	Simple file compression
Hardware	Enterprise hardware	Commodity hardware
Data access	Random access (indexing)	Access files only (streaming)
History	~40 years of innovation	< 5 years old
Community	Widely used, abundant resources	Not widely adopted yet

Figura 2.19 - Comparação entre RDBMS e Hadoop (Common, 2013)

2.4. WeDo Technologies

A WeDo Technologies é uma empresa líder mundial na área do Revenue Assurance e Fraud Management pertencente ao grupo Sonae, mais particularmente à SonaeIM. Foi fundada em fevereiro de 2001 tendo como cofundador e atual CEO Rui Paiva, realizou recentemente os seus 16 anos de existência. Inicialmente focou a sua atividade no mercado português, mas rapidamente a potencialidade do produto foi constatada fora de portas e a empresa procedeu à sua internacionalização. Tendo sido considerada, em junho de 2013, pelo Gartner o fornecedor de soluções de Revenue Assurance e Fraud Management na área das telecomunicações.

Orgulham-se de ter o software RAID, capaz de analisar dados ao longo da organização e melhorar a nível global a eficiência do negócio. Tem mais de 180 clientes em 108 países distintos e mais de 600 profissionais em mais de 10 escritórios ao longo do mundo (W. Technologies, 2018).

"We believe in a world where people make decisions based on real, truthful and clear digital information. Therefore a fair, transparent and sustainable world" (W. Technologies, 2018).

O software RAID foi totalmente desenvolvido pela empresa e é um tipo software de Business Intelligence que contém todas as ferramentas de Data Warehousing, ETL e Dashboarding necessárias à recolha de dados ao longo das plataformas de negócio para fornecer monotorização detalhada das atividades de negócio de modo a melhorar a performance.

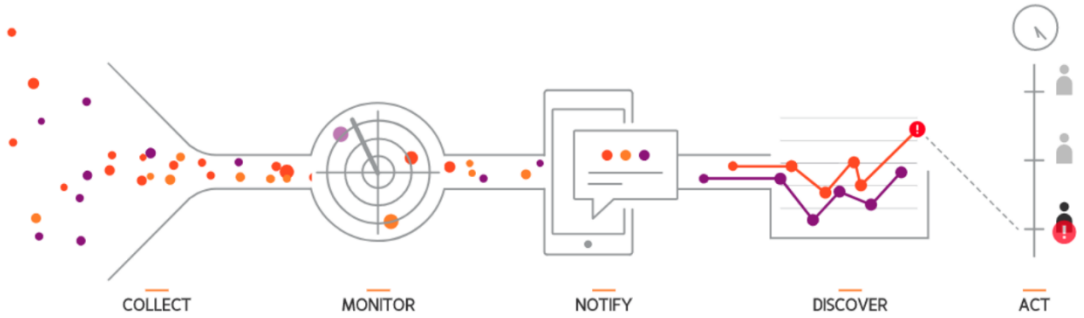


Figura 2.20 - Objetivo do software RAID (W. Technologies, 2018)

Tal como aplicado no software RAID, Ularu defende que "In telecom industry BI is used to monitor, analyze and provide key performance indicators (KPIs) on sales by product, region, distributor, partner, or sales representative using personalized BI dashboards and intuitive business intelligence reports" (Olaru, 2014).

Um dos fatores diferenciadores da WeDo para a concorrência, prende-se com o facto de os clientes poderem construir os seus próprios dashboards de forma simples, tal como poderiam fazê-lo numa ferramenta de Self-Service Business Intelligence (SSBI). O SSBI é definido como "end users designing and deploying their own reports and analyses within an approved and supported architecture and tools portfolio"(Gartner, 2017). Permitindo uma aproximação dos utilizadores do negócio ao software

de modo desenvolverem as suas próprias análises que poderão acrescentam valor ao negócio sem que estejam dependentes dos profissionais de IT.

Uma das lacunas que o software pretende responder prende-se com a dificuldade que as empresas de telecomunicação têm em compreender o seu cliente. Normile corrobora esta afirmação indicando que "Telecommunications vendors are rapidly acquiring significant product development capabilities as technology changes drive consumer demand. However, they continue to lag behind in understanding the customer" (Normile, 2011)

Num caso de estudo publicado pela WeDo Technologies (A. W. Technologies & Study, n.d.), estuda-se o modo como a empresa ajudou uma das grandes empresas da região asiática a monetizar o crescimento exponencial do consumo de dados. Serviu para o presente caso de estudo a empresa da região da asia com mais de 54,8 subescritores ativos na sua rede e que conta com mais 3 biliões de dados gerados por dia levando a cerca de 90 biliões de CDRs por mês e a abordagem de um processo switch-to-bill.

Para a implementação foram desenhadas duas fases. Numa primeira fase, a recolha de dados de doze elementos de Rede de modo a conseguir ter o controlo do fluxo de como os CDR's entram na rede. Numa fase posterior, procedeu-se ao desenvolvimento de 30 tipos de validações capazes de garantir que nenhuma transação não seria faturada.

Ainda no artigo a WeDo firma que o RAID permite que os fornecedores de serviços de telecomunicações se tornem mais proactivos e menos reativos na abordagem para com a empresa asiática. Tendo como benefícios:

- Processos automáticos altamente eficientes;
- Grande capacidade de armazenamento;
- Enriquecimentos de todo o processo prevenindo possíveis falhas na obtenção de lucros das empresas;
- Melhoramentos nas tomadas de decisões através de intuitivos dashboards, KPI's e capacidades de relatórios;

O RAID do ponto de vista da WeDo é o melhor software para tomadas de decisões baseadas no conhecimento. Trata-se de um software para recolher, analisar e correlacionar os dados tornando-os em informações valiosas para gestão de risco e fraude, em qualquer hora e em qualquer lugar (W. Technologies, 2018).

3. METODOLOGIA

Neste capítulo serão postos em prática os desenvolvimentos desde projeto conforme descrito anteriormente.

Numa primeira fase, após a recolha dos dados, será feita uma análise dos dados recolhidos assim como uma reflexão do modo que poderá se utilizar as funcionalidades do Software de Business Intelligence RAID.

Numa final desde capítulo, pretendo desenhar uma solução da arquitetura de como este novo método usado sobre o RAID poderá ser implementado e em que termos. De seguida, irá proceder-se a uma avaliação das queries e estruturas a serem usadas e de que forma serão efetuados os cálculos relativos a performance.

Para a execução deste projeto, a metodologia usada foi o "Design Science Research". Henver e Chatterjee definem o DSR como "research paradigm in which a designer answers questions relevant to human problems via the creation of innovative artifacts, thereby contributing new knowledge to the body of scientific evidence. The designed artifacts are both useful and fundamental in understanding that problem" afirmando no seu livro que um dos princípios fundamentais é que através da construção e a aplicação de artefactos é possível retirar conhecimento e compreensão para o desenho de soluções para os problemas (Hevner, Alan, Chatterjee, 2010).

Tendo em conta os diversos tipos de abordagens de pesquisas que existe segundo Järvinen, o que mais se aplica a este estudo será o "Artifacts Evaluating Approaches" que consiste em avaliar a efetividade do artefacto. O resultado final desta pesquisa será a medição da eficiência/eficácia de determinado artefacto que é avaliado através da utilização de determinado critério (Järvinen, 2000).

3.1. RECOLHA DE DADOS

Devido à política de privacidade e confidencialidade da WeDo Technologies para com os seus parceiros, não foi possível obter dados para a realização desde projeto.

Contudo, de acordo com o jornal internacional da ciência, Nature, uma das grandes empresas de telecomunicações de Itália disponibilizou online em 2015 um conjunto de dados composto por dados das áreas de telecomunicações, energia, mobilidade, turismo e fluxos de migração entre outras áreas na cidade de Milão e da Província de Trentino. Gianni Barlacchi e colegas afirmam que estes dados fornecidos pela empresa Italiana formam os testes ideais para metodologias e abordagens com o intuito de lidar com conjunto vasto problemas (Barlacchi et al., 2015).

De acordo com o mesmo artigo do parágrafo anterior, esta necessidade que a Telecom Itália teve em lançar um conjunto de dados Open Source deve-se ao facto do uso de telemóveis e o crescimento exponencial do uso de serviços de internet estarem a gerar quantidades enormes de dados que podem ser usados para fornecer novas perceções para sistemas técnico-sociais. Gianni afirma ainda que o facto de as empresas de telecomunicações não partilharem os seus dados pode trazer limitações para a comunidade científica tendo em conta as suas necessidades por potenciais estudos e criação de

problemas no processo de validação e reprodutibilidade. Neste seguimento, surgiu o “The Telecom Italia Big Data Challenge” (Barlacchi et al., 2015).

De acordo com o site da Telecom Itália, este desafio tem por objetivo estimular a inovação relacionada com o Big Data. Procurando pessoas em todo o mundo capazes de aceitar o desafio de desenvolver “the Big Data projects of the future” (Barlacchi et al., 2015).

Posto isto, de todo o conjunto de dados disponibilizado apenas irei usar os dados referentes às telecomunicações que já estão anonimizados.

3.2. CARACTERÍSTICAS

De acordo com o site Dandelion API (API, n.d.), de onde foram extraídos os dados. O conjunto de dados é composto pela computação de CDR’s gerado na província de Trento no norte italiano como demonstra a imagem abaixo.

Figura 3.1 - Província de Trento

O objetivo destes dados é controlar a atividade dos utilizadores, durante o mês de novembro de 2013, para posteriormente serem faturadas e para gestão de tráfego. De toda a panóplia de CDR’s apenas foram considerados pela empresa italiana:

- SMS Recebidos e Enviados
- Chamadas Recebidas e enviadas
- Internet (conexão à rede, a cada 15 minutos de utilização e a cada 5MB gerados)

Este conjunto de dados serve para medir o nível de interação entre os utilizadores e o seu telemóvel.

O conjunto de dados é composto por 4.3GB, distribuído por 30 ficheiros cada um corresponde a um dia. Tendo cada um desses ficheiros cerca de 2,5 milhões de registos.

Para o desenvolvimento deste projeto, visto que cada ficheiro tem, em média, cerca de 2 milhões de registos, irão ser utilizados os dois primeiros ficheiros:

- sms-call-internet-tn-2013-11-01.txt
- sms-call-internet-tn-2013-11-02.txt

Existem um total de 4 868 177 registos para análise. Os registos encontram-se compreendidos entre dia 31 de Novembro às 23Horas até ao dia 2 de Novembro até cerca das 22 horas.

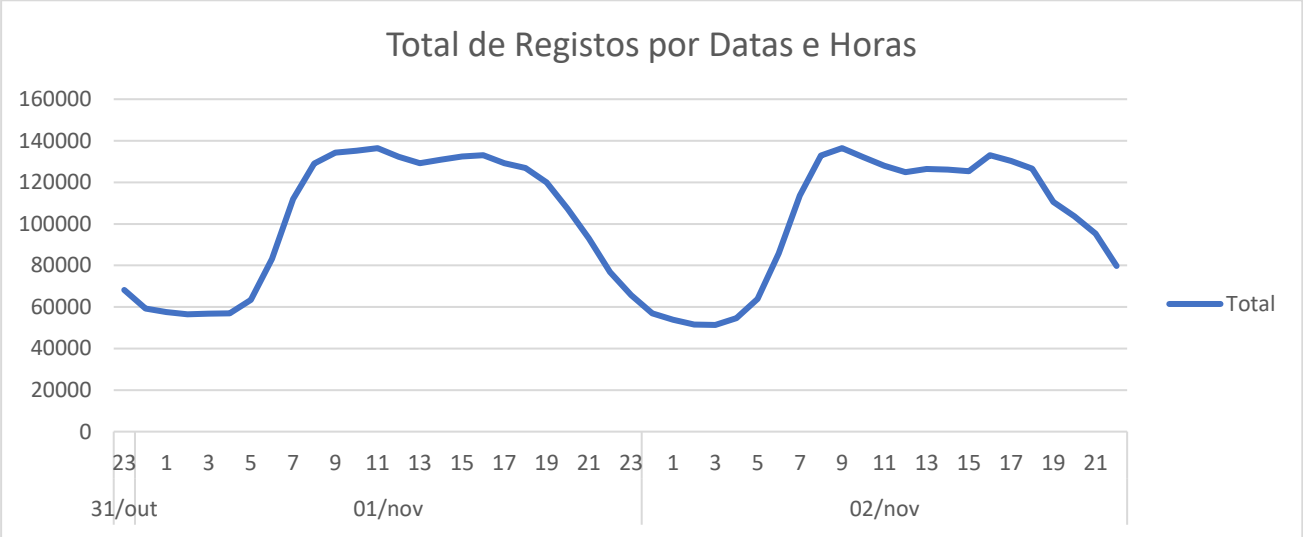


Figura 3.2 - Total de Registos por Datas e Horas

3.2.1. Atributos

A estrutura de cada ficheiro é composta por:

Tabela 3.1 - Descrição dos Dados

Tipo de Ficheiro:		.txt (Text File)		
Delimitador		\t (tabular)		
Coluna	Nome	Tipo	Descrição	Exemplo
1	Square id	Numérico (integer)	ID proveniente do conjunto de dados referente ao território da província de Trento	9999; 8808
2	Time interval	Numérico (integer)	Data expressa em milissegundos	1385302800000, 1385306400000
3	Country code	Numérico (integer)	Código telefónico de cada país.	40;39;0

4	SMS-in activity	Numérico (double)	Atividade de SMS recebidos associado a um ID espacial durante um determinado intervalo temporal e enviado a partir de um determinado país	0.075742408123834
5	SMS-out activity	Numérico (double)	Atividade de SMS enviados associado a um ID espacial durante um determinado intervalo temporal e enviado a partir de um determinado país	0.2990598185838835
6	Call-in activity	Numérico (double)	Atividade de Chamadas recebidas associado a um ID espacial durante um determinado intervalo temporal e enviado a partir de um determinado país	0.7243991736403472
7	Call-out activity	Numérico (double)	Atividade de Chamadas efetuadas associado a um ID espacial durante um determinado intervalo temporal e enviado a partir de um determinado país	0.6691270362806946
8	Internet traffic activity	Numérico (double)	Atividade de tráfego de internet face a um ID espacial durante um determinado intervalo temporal e enviado a partir de um determinado país	10.310409759677801

Nota: as atividades são obtidas pela agregação temporal dos CDR's em períodos de 10 em 10 minutos.

3.3. ELABORAÇÃO DA SOLUÇÃO

Tendo em vista o objetivo do estudo da performance da leitura dos dados em RAID através do Hadoop, numa fase inicial estudou-se como as tabelas são criadas nas respetivas conexões, Hive e Impala, os tipos de armazenamento possíveis e viáveis para cada uma delas e, por fim, os possíveis tipos de compressão.

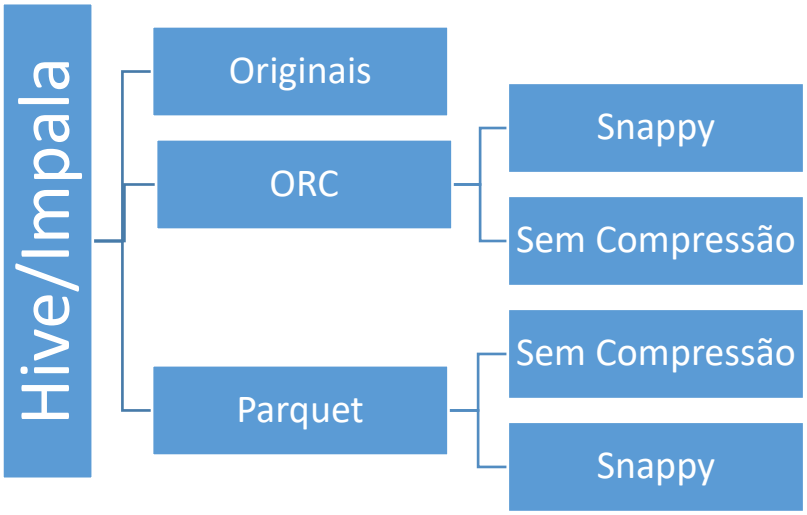


Figura 3.3 – Esquema dos tipos de armazenamento em Hadoop tendo em conta a sua compressão

Passada esta fase de avaliação, procedeu-se ao carregamento dos dados no HDFS e com a criação de várias pastas dedicadas a cada uma das possíveis combinações a que pertencerá cada uma das tabelas criadas.

O carregamento dos dados foi feito através de comandos Inserts. Após o carregamento, procedeu-se à elaboração das queries para a leitura dos dados.

3.4. AVALIAÇÃO INICIAL

Para conseguirmos obter uma comparação viável dos resultados será necessário replicar o processo que é considerado o habitual no RAID com os dados recolhidos, comparando os tempos de resultados em segundos das várias queries a serem executadas em Hive, Impala e Oracle.

Para isso a proposta de arquitetura a testar encontra-se representada na figura abaixo.

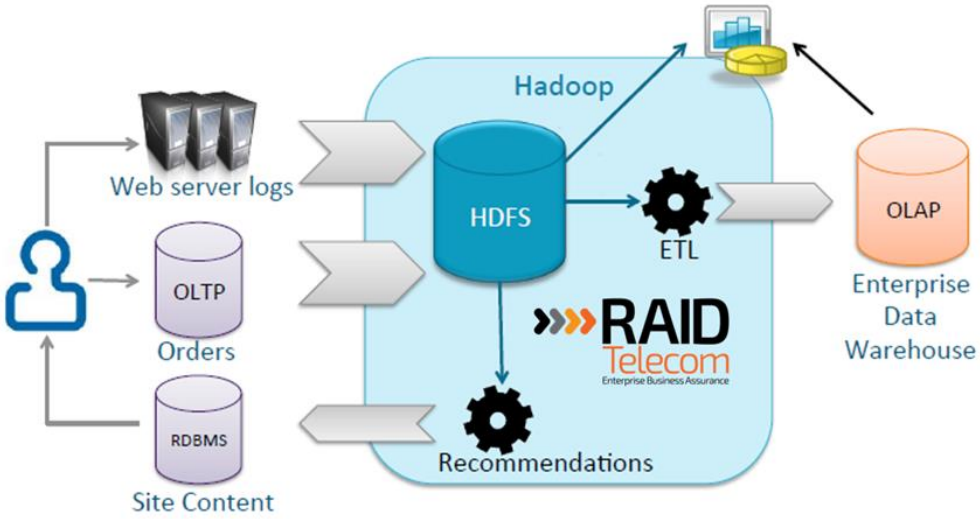


Figura 3.4 - Protótipo da Solução a Testar

3.4.1. Organização dos Dados

Nas áreas que se seguem teremos as estruturas das tabelas onde os dados serão armazenados tanto em Oracle como em Hadoop assim como a estrutura de diretórios onde foram colocados os ficheiros no RAID e no HDFS.

3.4.1.1. File System

Em termos do RAID, seguindo as práticas de usabilidade do software, os ficheiros foram colocados sobre a seguinte estrutura de diretórios:

`/projects/raid-prj/temp_data`

No HDFS, os mesmos ficheiros foram colocados na diretoria:

`/user/cloudera/test/`

- `original`
- `orc`
- `orc_snappy`
- `parquet`
- `parquet_snappy`

3.4.1.2. Hadoop

Com base no capítulo 2.2.1 acerca do ambiente do apache Hadoop em comparação ao Hive e ao Impala, seis tabelas foram criadas de modo a armazenar os dados tendo em conta as características em termos de tipos de formatos e métodos de compressão de cada um.

Tabela 3.2 - Descrição das Tabelas Criadas em Hadoop

Ligação	Nome Tabela	Descrição	Tipo de Tabela
Hive/Impala	original_table	Tabela comum em ambas as ligações onde os dados são armazenados na sua forma original sem qualquer tipo de formatação ou compressão.	External
Hive	orc_table	Dados Originais armazenados no formato Orc sem compressão especifica	External
	orc_snappy_table	Dados Originais armazenados no formato Orc com compressão to tipo Snappy	External
Impala	parquet_table	Dados Originais armazenados no formato Parquet sem compressão especifica	External
	parquet_snappy_table	Dados Originais armazenados no formato Parquet com compressão to tipo Snappy	External

Para a criação das tabelas, foi usado um comando comum onde apenas se muda o nome da tabela e os tipos de formatos e compressões que poderá ser consultado no Anexo 7.1:

3.4.1.3. Oracle

Para a organização dos dados em Oracle apenas uma tabela foi criada para armazenamento dos dados originais (Anexo 7.1).

Contudo, outra tabela foi criada para registo dos tempos e estados de execução de cada fluxo de modo a tornar este processo mais autónomo e preciso na obtenção de resultados de forma mais rápida. Essa tabela é composta pela seguinte informação:

Tabela 3.3 - Descrição da Tabela de Controlo

Nome da Tabela	Colunas	Tipo	Descrição
control_execution_table	querie_id	Integer	Número da querie que foi executada
	start_date_execution	Date	Data do inicio da execução do tipo yyyy-mm-dd hh24:mi:ss
	end_date_execution	Date	Data do fim da execução do tipo yyyy-mm-dd hh24:mi:ss
	execution_number	Integer	Número da execução
	table_name	String	Nome da tabela onde a querie foi executada
	connection_type	String	Tipo de conexão. Exemplo: Hive, Impala ou Oracle
	flow_execution_id	Integer	Número de execução gerado automaticamente pelo fluxo

3.4.2. Construção das Queries

Como já foi abordado nos capítulos anteriores, a existência de artigos referentes à performance entre o Hive e o Impala ou até mesmo entre o Hive e o Oracle é vasta assim como os exemplos de queries usadas para o estudo desses mesmos artigos. Contudo, pretende-se combinar todos esses exemplos já existentes e testar a leitura dos dados propostos no capítulo 3.1 no software RAID.

Para isso foi criada uma matriz com base no artigo de Floratou e a sua equipa da IBM quando estudaram a performance entre o Hive e o Impala (Floratou et al., 2014) que será preenchida com base na informação obtida:

Tabela 3.4 - Descrição da Matriz

Colunas		Descrição da Coluna
Queries		Número da Querie
Execuções		Número de Execuções de 1 a 10
Hive	**TXT**	Tempo de execução em segundos dos dados em Hive no formato Original (.txt)
	ORC	Tempo de execução em segundos dos dados em Hive no formato colunar Orc
	ORC Snappy	Tempo de execução em segundos dos dados em Hive no formato colunar Orc usando o método de compressão Snappy

Colunas		Descrição da Coluna
Impala	**TXT**	Tempo de execução em segundos dos dados em Impala no formato Original (.txt)
	Parquet	Tempo de execução em segundos dos dados em Impala no formato colunar Parquet
	Parquet Snappy	Tempo de execução em segundos dos dados em Impala no formato colunar Parquet usando o método de compressão Snappy
Oracle		Tempo de execução em segundos dos dados em Sql Developer
Média		Média dos tempos de execução

A construção das queries foi baseada no paper escrito por Sethy do departamento de ciências computacionais da universidade de Utkal na India (Sethy et al., 2018). As queries a serem executadas serão:

Querie 1:

Tem como objetivo analisar o comportamento de ambos os softwares quando lhes é mandado pesquisar todos os dados existentes nas tabelas tendo de retornar os 4 868 177 registos:

```
SELECT SQUARE_ID,
  SMS_IN_ACTIVITY,
  SMS_OUT_ACTIVITY,
  INTERNET_TRAFFIC_ACTIVITY,
  CALL_OUT_ACTIVITY,
  CALL_IN_ACTIVITY,
  TIME_INTERVAL,
  COUNTRY_CODE
FROM <Nome_Tabela>;
```

Querie 2:

Perceber os tempos que as plataformas demoram a devolver os valores únicos da coluna em estudo.

```
SELECT DISTINCT COUNTRY_CODE
FROM <Nome_Tabela>;
```

Tendo de retornar 194 registos.

Querie 3:

Avaliar quanto tempo demora a realizar a ordenação dos dados por datas:

```
SELECT SQUARE_ID,
  SMS_IN_ACTIVITY,
  SMS_OUT_ACTIVITY,
  INTERNET_TRAFFIC_ACTIVITY,
```

```
  CALL_OUT_ACTIVITY,
  CALL_IN_ACTIVITY,
  TIME_INTERVAL,
  COUNTRY_CODE
FROM <Nome_Tabela>
ORDER BY TIME_INTERVAL ASC;
```

Retorna os 4 868 177 registos ordenados por data de forma ascendente.

Querie 4:

Analisar os tempos usando uma clausula Where:

```
SELECT SQUARE_ID,
  SMS_IN_ACTIVITY,
  SMS_OUT_ACTIVITY,
  INTERNET_TRAFFIC_ACTIVITY,
  CALL_OUT_ACTIVITY,
  CALL_IN_ACTIVITY,
  TIME_INTERVAL,
  COUNTRY_CODE
FROM <Nome_Tabela>
WHERE COUNTRY_CODE = '351';
```

Retorna 6 749 registos.

Querie 5:

Avaliar a performance usando medidas de agregação:

```
SELECT count(*) FROM <Nome_Tabela>;
```

Retorna 1 registo.

Querie 6:

Avaliar a performance usando medidas de agregação através da agregação dos dados:

```
SELECT TIME_INTERVAL, count(*)
FROM <Nome_Tabela>
GROUP BY TIME_INTERVAL;
```

Retorna 288 registos.

Para o processo de seleção de queries para comparação de performance com o Oracle será utilizado o seguinte critério:

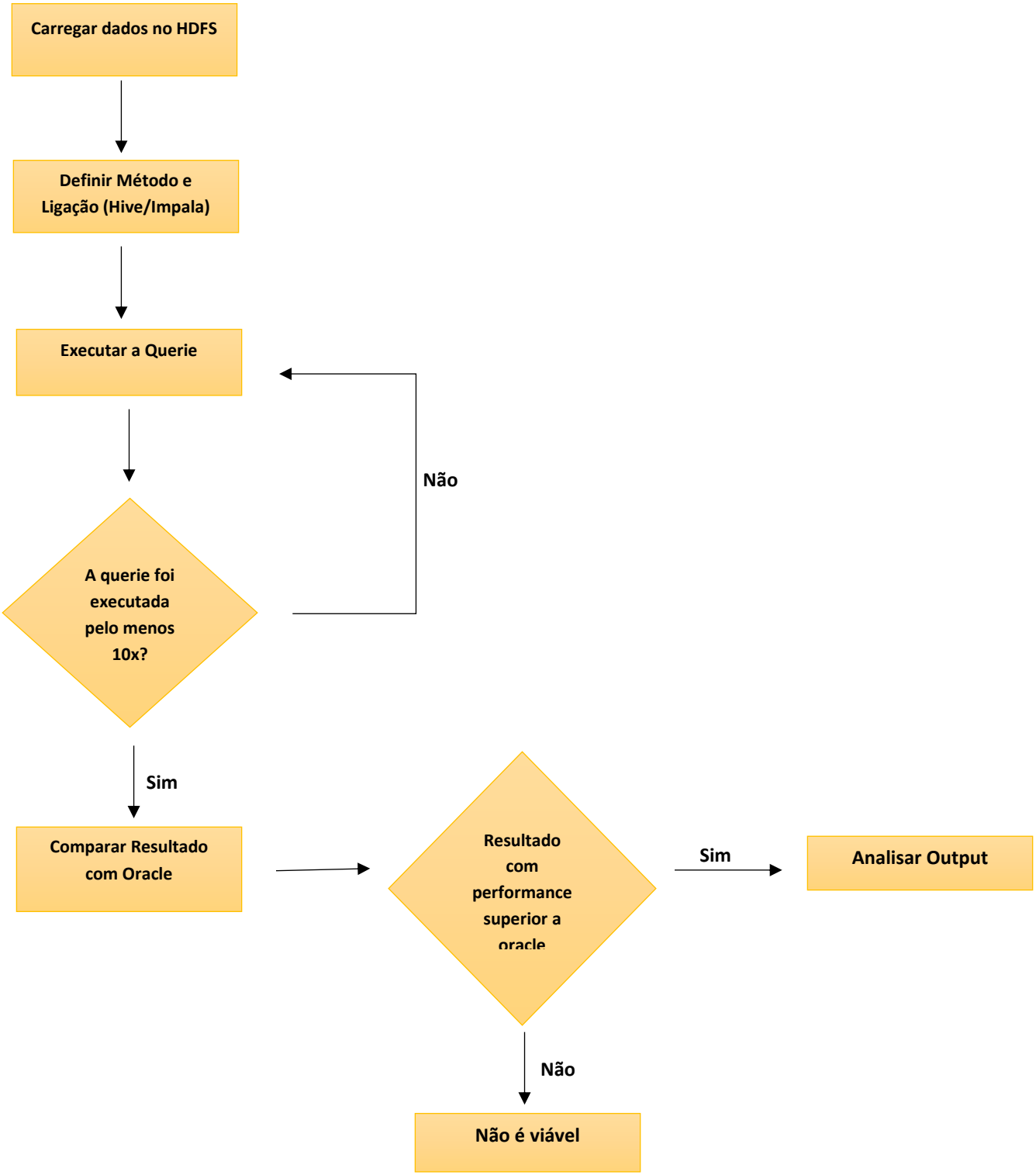


Figura 3.5 – Processo de Avaliação dos Resultados

4. RESULTADOS

Neste capítulo irei inicialmente analisar o espaço ocupados pelos dados no HDFS de modo a comparar os resultados obtidos com o que foi descrito na Revisão de Literatura, tendo em conta os tipos de ficheiros e os métodos de compressão.

De seguida passarei para as análises dos resultados obtidos e apresentados no anexo 7.2. Será feita a análise individual dos tempos de resposta do software RAID na leitura dos dados de cada uma das queries propostas anteriormente no capítulo 3.4.2, comparativamente ás tabelas em cada tipo de conexão usada.

Irei também comparar os tempos de resposta da leitura dos dados tendo em conta apenas os tipos de conexão Oracle vs. Hive e Oracle vs. Impala.

4.1. ANÁLISE DO ESPAÇO NO HDFS VS LINUX FILE SYSTEM

Os ficheiros a serem analisados ocuparão cerca de 328Mb no sistema de repositório de ficheiros. Este é utilizado atualmente nas soluções da WeDo Technologies conforme demostrado na tabela abaixo.

Tabela 4.1 - Espaço Ocupado Pelos Ficheiros no Repositório Linux

Nome do Ficheiro	**Espaço Ocupado (Mb)**
sms-call-internet-tn-2013-11-01.txt	164
sms-call-internet-tn-2013-11-02.txt	160
Total	**328**

Conforme descrito na revisão de literatura, no capítulo referente ao Apache Hadoop (2.2.1), a figura 4.1, comprova que os ficheiros escritos em no formato colunar ORC tornam-se mais eficientes na poupança de espaço e na resolução do problema em servidores obsoletos.

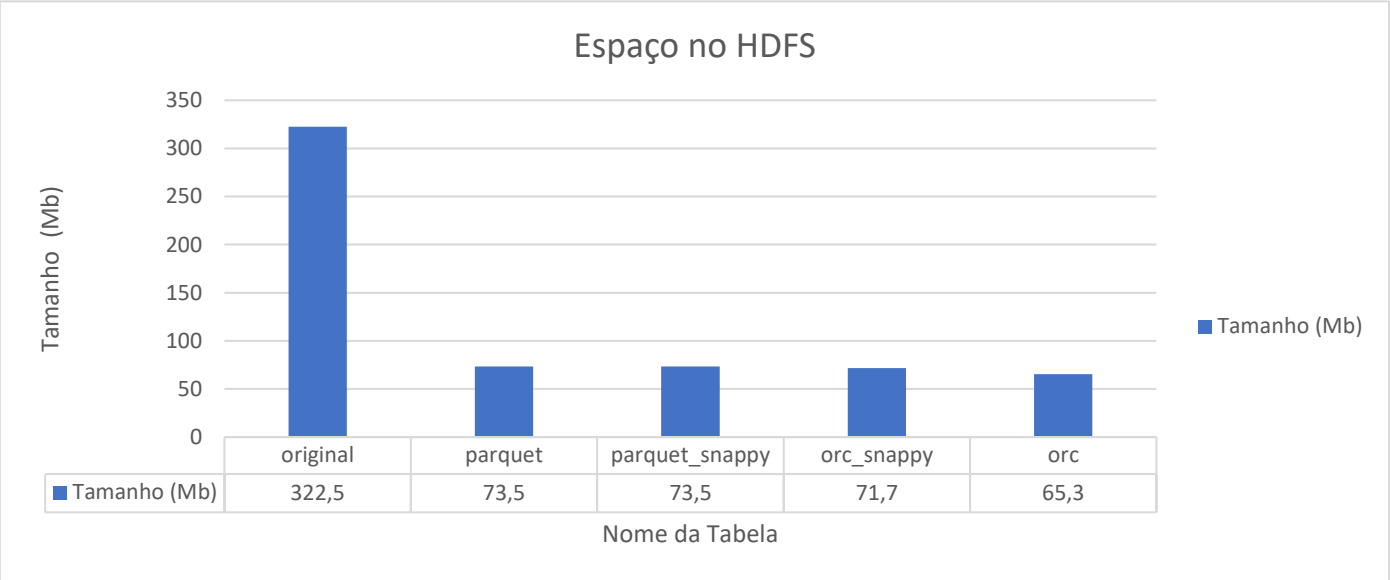


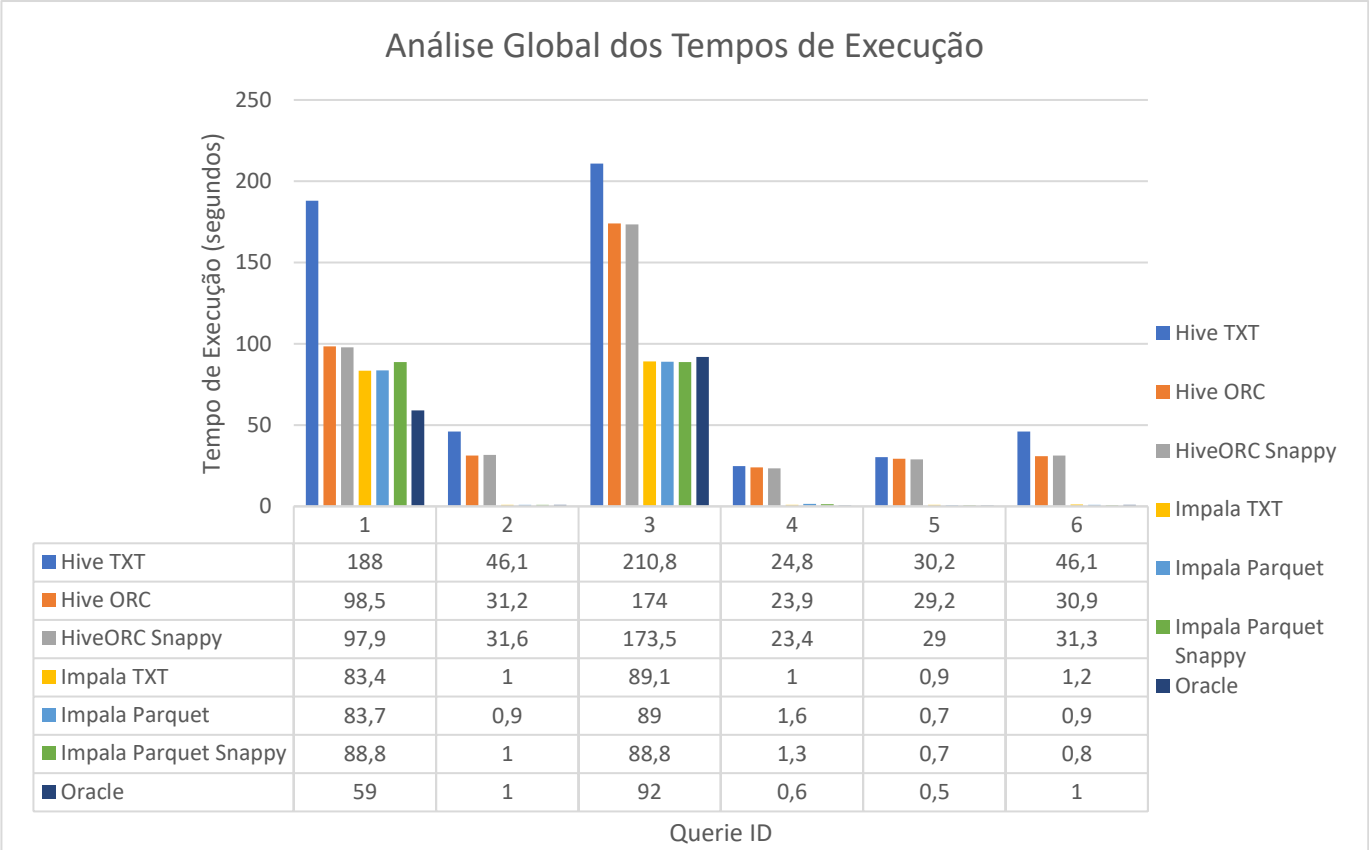
Figura 4.1 - Espaço Ocupado no HDFS

O pouco espaço no filesystem não significa melhor performance na execução de queries, porém poderá ajudar. Ao existir mais espaço disponível no HDFS para execução das queries poderá torna-las mais rápidas.

4.2. Análise das Execuções

Apesar do que foi descrito no processo de seleção representado na figura 3.5, onde apenas se iria comparar os resultados que tivessem performance superior a Oracle, tal não foi possível de se realizar. Uma vez que no global nenhuma execução foi superior á performance que o RAID tem de leitura de dados em Oracle. A que mais se aproximou foi a conexão realizada com o Impala. Contudo, as análises com o Hive não foram completamente descartadas, pois será interessante perceber a disparidade dos resultados obtidos.

Numa primeira análise, estão descritos na figura 4.2 os tempos médios das dez execuções tendo em contas os diferentes tipos de dados analisados. É possível se aperceber que as execuções do Hive são as que têm os piores tempos. Têm em média uma performance vinte e nove vezes mais lenta que a solução em Oracle. Por outro lado, na generalidade dos casos o Impala assemelha-se muito aos resultados do Oracle. No entanto, em média é 1,34 segundos mais lento o que poderá não ser significativo.



	1	2	3	4	5	6
Hive TXT	188	46,1	210,8	24,8	30,2	46,1
Hive ORC	98,5	31,2	174	23,9	29,2	30,9
HiveORC Snappy	97,9	31,6	173,5	23,4	29	31,3
Impala TXT	83,4	1	89,1	1	0,9	1,2
Impala Parquet	83,7	0,9	89	1,6	0,7	0,9
Impala Parquet Snappy	88,8	1	88,8	1,3	0,7	0,8
Oracle	59	1	92	0,6	0,5	1

Figura 4.2 - Análise Global dos Tempos de Execução

Analisando o Output de todas as Queries individualmente (Anexo 7.3) é possível perceber que realizar qualquer tipo de análises sobre a tabela onde os dados estão inseridos sem qualquer tipo de compressão ou formato específico não é viável. Sendo cerca de 354% mais lento do que o Oracle onde a pior execução foi a análise referente à ordenação do conjunto de dados.

Os outros tipos de tabelas ligadas ao Hive, embora tenham melhores resultados comparativamente à tabela original são, em média, 250% mais lentos que o Oracle.

Muitos dos resultados que se obtiveram em relação ao Hive devem-se aos tempos de demora nos processos de Map e Reduce.

Por outro lado, o Impala exceto na Querie 1 é o que mais se assemelha ao resultado apresentado em Oracle no que toca à organização dos registos. É igualmente eficaz na execução de análises que requerem agregações dos dados em comparação com Oracle.

O impala consegue ser mais rápido, devido ao facto de em primeiro lugar colocar os dados em memória. Tendo os dados em memoria não se irá consumir CPU e RAM à infraestrutura.

Comparando a média dos resultados de todas as execuções de cada tipo de conexão os resultados obtidos foram:

Tabela 4.2 - Média de todas as conexões por Tipos de Conexões

Nome da Tabela por Tipo de Conexão	Média de Todas as Execuções (segundos)
Hive TXT	91
Hive ORC	64,62
Hive ORC Snappy	64,45
Impala TXT	29,43
Impala Parquet	29,47
Impala Parquet Snappy	30,23
Oracle	25,68

5. CONCLUSÕES

Tendo em conta os resultados obtidos no capítulo anterior, as conclusões que se poderão obter são várias consoante os diferentes cenários possíveis.

É visível a discrepância que existe em termos de tempos para todas as hipóteses geradas entre o Hive e o Oracle o que nos leva a concluir que para o Software da WeDo Technologies a aposta será vender a solução que existe atualmente na empresa. Por outro lado, isto acontece devido á otimização dos conectores e ligações entre o Oracle e o RAID. Pelo que poderei sugerir otimização dos mesmo para Hive, em concreto para processos de MapReduce e re-testar a solução toda para versões futuras.

Relativamente à comparação do Oracle com o Impala, embora existam casos em que o resultado de ambos seja semelhante, é possível afirmar que o Impala se pode tornar uma solução viável. Existindo a limitação do espaço em memória à qual se procedeu alguns ajustes na execução do projeto para não se chegar ao limite da mesma.

Apesar de alguns ajustes necessários na solução, a aposta na área de Big Data por parte da WeDo Technologies, visto ser uma empresa líder de mercado, deve ser considerada devido ao crescimento exponencial que tem tido como ficou comprovado ao longo do capítulo de Revisão de Literatura.

De modo a seguir com esta evolução dentro da empresa o modelo de solução proposto é facilmente aplicável nos testes de performance de versões de software futuras. Devendo apostar-se na existência de mais conhecimento e otimização para compatibilidade de ambas as soluções Hadoop e Oracle.

Analisando o Resultado final do projeto, conclui-se que para já a solução que trará mais performance ao Software RAID será a tradicional, não obstante á existência de projetos em Hadoop. Estes poderão servir de aprendizagem para o futuro e para entrada no mercado na área. Servindo para identificar oportunidades existentes de desenvolvimento de novas soluções do RAID, de modo a torna-lo mais competitivo.

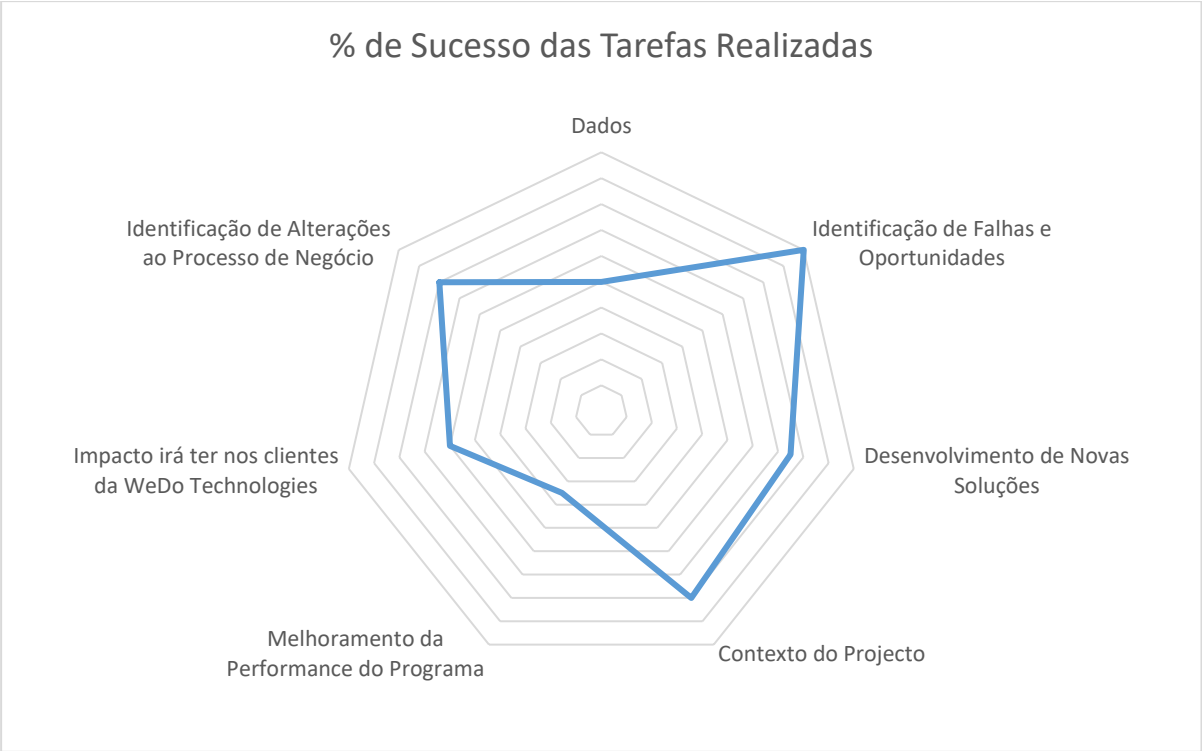


Figura 5.1 - Percentagem de Sucesso das Tarefas Realizadas

Embora o resultado não tenha sido o esperado, devido à espectativa criada durante a revisão de literatura sinto que foram cumpridos todos os objetivos propostos. Na figura 5.1, sugere-se um pequeno sumário do que foi atingido na realização deste projeto tendo em conta os objetivos.

5.1.1. Limitações do Projeto

Durante o desenvolvimento deste projeto, as limitações encontradas foram:

- Falta de dados devido á nova lei de proteção de dados por parte da empresa WeDo Techonologies. Tendo sido necessário recorrer a dados externos considerados Open Data.
- Alguns problemas de recursos dos servidores que levavam a algum desacelaramento no processo de obtenção de resultados e algumas paragens, pois era necessária a intervenção para o aumento de memória da máquina onde foi realizada o projeto.
- A abertura de vários problemas á equipa encarregue do desenvolvimento do produto, devido a adversidades e falhas de funcionalidade existentes no decorrer deste trabalho, o que apenas se torna uma limitação pelo tempo de resolução dos problemas e buscas de justificação para o comportamento irregular de algumas funcionalidades do RAID. Pois, o facto de estes mesmo problemas terem sido reportados ajudou contribuir para o melhoramento do produto.

5.1.2. Recomendações Futuras

Em termos de recomendações futuras, gostava de salientar os seguintes aspetos:

- Como ficou comprovado nos capítulos anteriores, da mesma maneira que o produto tem as queries geradas em Oracle automatizadas também se devia fazer o mesmo para as queries HiveQL or Impala.
- Seria interessante a existência de um mecanismo de conversão de queries consoante o tipo de conexão à linguagem SQL.
- No pacote de instalação do produto, a estrutura de diretórios onde se devem colocar os drives e os drives base (exceto as nativas do Hive ou impala que podem variar de cliente para cliente consoante os que têm na distribuição dos seus HIVE clusters) já deviam vir no momento de instalação.
- Visto que neste momento ainda são poucos os projetos dentro da empresa que utilizam Hadoop, mas o crescimento é exponencial dever-se-ia proceder à realização de documentação oficial para a instalação referida no ponto anterior.

Contudo, para além de recomendações ainda existe espaço para continuar a explorar este tema.

- Para além da leitura e apresentação dos dados, existe espaço para explorar o modo como o software se irá comportar durante o processo de ETL. Contudo, este ponto poderá não ter grande impacto, pois sempre que o RAID lê os dados estes ficam alocados na memória do Software.
- Escrita em Hadoop e gestão dos ficheiros em HDFS como processos de limpeza (processos de Houseekeeping)

6. REFERÊNCIAS

apache. (n.d.). Apache ORC.

API, D. (n.d.). No Title.

Barclay, M. (2015). Business Metrics vs. KPI's – The Difference. Retrieved from http://datapathfinders.com/business-metrics-kpis-the-difference/

Barlacchi, G., De Nadai, M., Larcher, R., Casella, A., Chitic, C., Torrisi, G., … Lepri, B. (2015). A multi-source dataset of urban life in the city of Milan and the Province of Trentino. *Scientific Data*, *2*, 1–15. https://doi.org/10.1038/sdata.2015.55

Castellanosm M., D. U. (2008). *Business Intelligence for the Real-Time Enterprise*.

Chaudhuri, B. Y. S., Dayal, U., & Narasayya, V. (2011). BI-Tech2. https://doi.org/10.1145/1978542.1978562

Chen, H., & Storey, V. C. (2012). B Usiness I Ntelligence and a Nalytics : F Rom B Ig D Ata To B Ig I Mpact. *Mis Quarterly*, *36*(4), 1165–1188. https://doi.org/10.1145/2463676.2463712

Cloudera. (n.d.). CDH Overview. Retrieved from https://www.cloudera.com/documentation/enterprise/5-7-x/topics/cdh_intro.html

Common, H. (2013). Introduction to Hadoop What is Distributed Computing ? What is Hadoop ?

Dijcks, J. (2012). Oracle: Big data for the enterprise. *Oracle White Paper*, (June), 16. Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Oracle+:+Big+Data+for+the+Enterprise#0

Dumbill. (2012). What is Big Data? In: O'Reilly Media Inc. Big Data Now: current perspectives. Retrieved from http://www.oreilly.com/data/free/files/big%0A-data-now-2012.pdf

Floratou, A., Minhas, U. F., & Ozcan, F. (2014). Sql-on-hadoop: Full circle back to shared-nothing database architectures. *Proceedings of the VLDB Endowment*, *7*(12), 1295–1306. https://doi.org/10.14778/2732977.2733002

Gantz, J., & Reinsel, D. (2012). THE DIGITAL UNIVERSE IN 2020: Big Data, Bigger Digi tal Shadows, and Biggest Growth in the Far East. *Idc*, *2007*(December 2012), 1–16.

Gartner. (2017). Gartner IT Glossary. Retrieved from https://www.gartner.com/it-glossary/self-service-business-intelligence

Ghazi, M. R., & Gangodkar, D. (2015). Hadoop, mapreduce and HDFS: A developers perspective. *Procedia Computer Science*, *48*(C), 45–50. https://doi.org/10.1016/j.procs.2015.04.108

Goldman, A., Kon, F., Junior, F. P., Polato, I., & Pereira, R. de F. (2012). Apache Hadoop: conceitos teóricos e práticos, evolução e novas possibilidades. *VII Jornadas de Atualização Em Informática*, 86–136.

Hadoop. (n.d.). HDFS Architecture Guide. Retrieved from http://hadoop.apache.org

Hevner, Alan, Chatterjee, S. (2010). *Design Research in Information Systems - Theory and Practice (Vol. 22)*. (Springer, Ed.). Retrieved from http://www.springer.com/business+%26+management/business+information+systems/book/9

78-1-4419-5652-1

Hive. (2018). Apache Hive. Retrieved from http://hive.apache.org/

Hortonworks. (n.d.). ORCFile in HDP 2: Better Compression, Better Performance.

Impala, A. (n.d.). No Title. Retrieved from https://impala.apache.org

Järvinen, P. (2000). Research questions guiding selection of an appropriate research method. *ECIS Proceedings*, *3*(5.6), 124–131. https://doi.org/10.1.1.144.2055

Kornacker, M., Behm, A., Bittorf, V., Bobrovytsky, T., Russell, J., Tsirogiannis, D., & Michael, S. W. (n.d.). Impala : A Modern , Open-Source SQL Engine for Hadoop.

Kumar, P. (2012). Impact of Business Intelligence systems in Indian Telecom Industry. *Business Intelligence Journal*, *5*(2), 358–366.

Laney, D. (2001). META Delta. *Application Delivery Strategies*, *949*(February 2001), 4. https://doi.org/10.1016/j.infsof.2008.09.005

McKinsey & Company. (2011). Big data: The next frontier for innovation, competition, and productivity. *McKinsey Global Institute*, (June), 156. https://doi.org/10.1080/01443610903114527

Neely, A., Gregory, M., & Platts, K. (2005). Performance measurement system design: A literature review and research agenda. *International Journal of Operations and Production Management*, *25*(12), 1228–1263. https://doi.org/10.1108/01443570510633639

Negash, S. (2004). Business Intelligence. *Communications of the Association for Information Systems*, *Vol. 13*(February), Article 15. https://doi.org/10.1002/9781118915240.ch7

Normile, C. (2011). BusIness IntellIgence for the telecommunIcatIons Industry. *Ingress*.

Olaru, C. (2014). Business Intelligence in Telecommunications Industry. *International Journal of Economic Practices and Theories*, *4*(1), 89–100.

Ribeiro, C. J. S. (2014). Big Data: os novos desafios para o profissional da informação. *Informação & Tecnologia*, *1*(1), 96–105.

Roth, E. (2017). How to Define KPIs for Successful Business Intelligence. Retrieved from https://www.sisense.com/blog/how-to-define-kpis-for-successful-business-intelligence/

Rouse, M. (2015). What is Business Metric? Retrieved from http://searchcrm.techtarget.com/definition/business-metric

Saraladevi, B., Pazhaniraja, N., Paul, P. V., Basha, M. S. S., & Dhavachelvan, P. (2015). Big data and Hadoop-A study in security perspective. *Procedia Computer Science*, *50*, 596–601. https://doi.org/10.1016/j.procs.2015.04.091

SAS. (n.d.). What is Big Data?

Science, N. D. of P. C. H. (n.d.). Big Data. Retrieved from https://www.phc.ox.ac.uk/research/big-data

Sethy, R., Dash, S. K., & Panda, M. (2018). Proceedings of the Eighth International Conference on Soft Computing and Pattern Recognition (SoCPaR 2016), *614*(October). https://doi.org/10.1007/978-3-319-60618-7

Spark. (2018). Apache Spark. Retrieved from http://spark.apache.org/

Sqoop. (2018). Apache Sqoop. Retrieved from http://sqoop.apache.org/

Stage, A. R. (2006). Techniques, Process, and Enterprise Solutions of Business Intelligence, 4722–4726.

Technologies, A. W., & Study, C. (n.d.). Leveraging AN INNOVATIVE MOBILE DATA OPERATOR IN ASIA into new levels of control and monitoring capabilities Maximize the value creation How Wedo Technologies is helping a high-speed mobile operator in monetizing.

Technologies, W. (2018). WeDo Technologies. Retrieved from WeDo Technologies

Thusoo, A., Sarma, J. Sen, Jain, N., Shao, Z., Chakka, P., Anthony, S., … Murthy, R. (2009). Hive - A Warehousing Solution Over a Map-Reduce Framework. *Sort*, *2*, 1626–1629. https://doi.org/10.1109/ICDE.2010.5447738

Thusoo, A., Sarma, J. Sen, Jain, N., Shao, Z., Chakka, P., Zhang, N., … Murthy, R. (2010). Hive - A petabyte scale data warehouse using hadoop. *Proceedings - International Conference on Data Engineering*, 996–1005. https://doi.org/10.1109/ICDE.2010.5447738

Villars, R. L., & Olofson, C. W. (2014). WHITE P APER Big Data : What It Is and Why You Should Care INFORMATION EVERYWHE RE , BUT WHERE ' S THE KNOWLEDGE ?

Wang, Y., Xu, Y., Liu, Y., Chen, J., & Hu, S. (2015). QMapper for Smart Grid : Migrating SQL-based Application to Hive QMapper for Smart Grid : Migrating SQL-based Application to Hive, (January).

Ward, J. S., & Barker, A. (2013). Undefined By Data: A Survey of Big Data Definitions. https://doi.org/10.1145/2699414

Webb, S. A. M. C. (2015). Report-Writing Ability., 31–46. https://doi.org/10.1145/248603.248616

White, T. (2010). *Hadoop : the definitive guide*.

7. ANEXOS

7.1. CÓDIGO DE CRIAÇÃO DAS TABELAS EM HIVE, IMPALA E ORACLE

Tabelas Originais:

```
Create external table original_table
(square_id int,
 time_interval int,
 country_code int,
 sms_in_activity double,
 sms_out_activity double,
 call_in_activity double,
 call_out_activity double,
 internet_traffic_activity double
)
row format delimited
fields terminated by '\t'
location '/user/cloudera/test/original';
```

Tabela Orc:

```
Create external table orc_table
(square_id int,
 time_interval int,
 country_code int,
 sms_in_activity double,
 sms_out_activity double,
 call_in_activity double,
 call_out_activity double,
 internet_traffic_activity double
)
row format delimited
fields terminated by '\t'
stored as ORC
location '/user/cloudera/test/orc';
```

Tabela Orc com compressão Snappy:

```
Create external table orc_snappy_table
(square_id int,
 time_interval int,
 country_code int,
 sms_in_activity double,
 sms_out_activity double,
 call_in_activity double,
 call_out_activity double,
 internet_traffic_activity double
)
row format delimited
fields terminated by '\t'
stored as ORC
```

```
location '/user/cloudera/test/orc_snappy'
tblproperties ("orc.compress"="snappy");
```

Tabela Parquet:

```
Create external table parquet_table
(square_id int,
 time_interval int,
 country_code int,
 sms_in_activity double,
 sms_out_activity double,
 call_in_activity double,
 call_out_activity double,
 internet_traffic_activity double
)
row format delimited
fields terminated by '\t'
stored as PARQUET
location '/user/cloudera/test/parquet';
```

Tabela Parquet com compressão Snappy:

```
Create external table parquet_snappy_table
(square_id int,
 time_interval int,
 country_code int,
 sms_in_activity double,
 sms_out_activity double,
 call_in_activity double,
 call_out_activity double,
 internet_traffic_activity double
)
row format delimited
fields terminated by '\t'
stored as PARQUET
location '/user/cloudera/test/parquet_snappy'
tblproperties ("parquet.compress"="snappy");
```

Oracle:

```
Create table original_table
(square_id number(10,0),
 time_interval number(10,0),
 country_code number(10,0),
 sms_in_activity number(10,20),
 sms_out_activity number(10,20),
 call_in_activity number(10,20),
 call_out_activity number(10,20),
 internet_traffic_activity number(10,20)
);
```

7.2. MATRIZ DOS RESULTADOS PREENCHIDA

Resultados expressos em Segundos

Tabela 7.1 - Matriz dos Resultados Preenchida

Queries	Execuções	Hive			Impala			Oracle	Média
		TXT	ORC	ORC Snappy	TXT	Parquet	Parquet Snappy		
1	1	186	99	99	87	88	60	58	96,7
	2	187	99	98	84	82	165	59	110,6
	3	187	98	98	83	84	82	59	98,7
	4	189	98	97	82	83	82	59	98,6
	5	189	98	98	83	83	83	60	99,1
	6	186	98	98	84	83	84	58	98,7
	7	187	99	100	82	84	83	59	99,1
	8	192	99	97	83	82	81	59	99,0
	9	188	99	97	83	84	84	59	99,1
	10	189	98	97	83	84	84	60	99,3
2	1	47	32	32	1	1	1	1	16,4
	2	44	32	31	1	1	1	1	15,9
	3	46	30	32	1	1	1	1	16,0
	4	47	31	32	1	1	1	1	16,3
	5	45	32	31	1	1	1	1	16,0
	6	47	31	32	1	1	1	1	16,3
	7	47	32	32	1	1	1	1	16,4
	8	46	30	32	1	1	1	1	16,0
	9	46	32	31	1	0	1	1	16,0
	10	46	30	31	1	1	1	1	15,9
3	1	212	174	174	90	89	88	60	126,7

Queries	Execuções	Hive			Impala			Oracle	Média
		TXT	ORC	ORC Snappy	TXT	Parquet	Parquet Snappy		
	2	214	175	173	90	90	88	60	127,1
	3	210	174	172	88	88	88	238	151,1
	4	209	173	173	88	88	89	81	128,7
	5	209	175	175	90	89	90	80	129,7
	6	209	171	172	90	90	89	80	128,7
	7	210	172	173	88	90	89	80	128,9
	8	211	175	174	88	88	88	80	129,1
	9	212	177	176	90	88	90	80	130,4
	10	212	174	173	89	90	89	81	129,7
4	1	26	23	24	1	2	1	0	11,0
	2	24	25	22	1	1	2	1	10,9
	3	25	24	25	1	1	2	0	11,1
	4	24	24	23	1	2	1	0	10,7
	5	25	23	23	1	1	1	1	10,7
	6	24	25	23	1	2	1	1	11,0
	7	25	24	24	1	2	1	1	11,1
	8	25	23	24	1	1	2	1	11,0
	9	25	24	23	1	2	1	0	10,9
	10	25	24	23	1	2	1	1	11,0
5	1	28	28	28	1	1	1	1	12,6
	2	33	28	30	1	0	1	0	13,3
	3	29	33	29	0	0	1	1	13,3
	4	30	29	30	1	1	0	0	13,0
	5	29	29	28	1	1	1	1	12,9

Queries	Execuções	Hive			Impala			Oracle	Média
		TXT	ORC	ORC Snappy	TXT	Parquet	Parquet Snappy		
	6	31	29	30	1	1	1	0	13,3
	7	30	29	29	1	1	0	0	12,9
	8	30	29	29	1	1	1	1	13,1
	9	31	29	28	1	0	1	0	12,9
	10	31	29	29	1	1	0	1	13,1
6	1	44	31	31	2	1	1	1	15,9
	2	46	30	32	1	1	1	1	16,0
	3	47	32	32	1	1	1	1	16,4
	4	47	31	31	1	1	1	1	16,1
	5	46	31	31	1	1	1	1	16,0
	6	46	30	33	1	1	0	1	16,0
	7	48	30	31	1	1	1	1	16,1
	8	46	31	31	1	1	1	1	16,0
	9	45	30	31	1	1	1	1	15,7
	10	46	33	30	2	0	0	1	16,0

7.3. GRÁFICOS DOS RESULTADOS

7.3.1. Querie 1

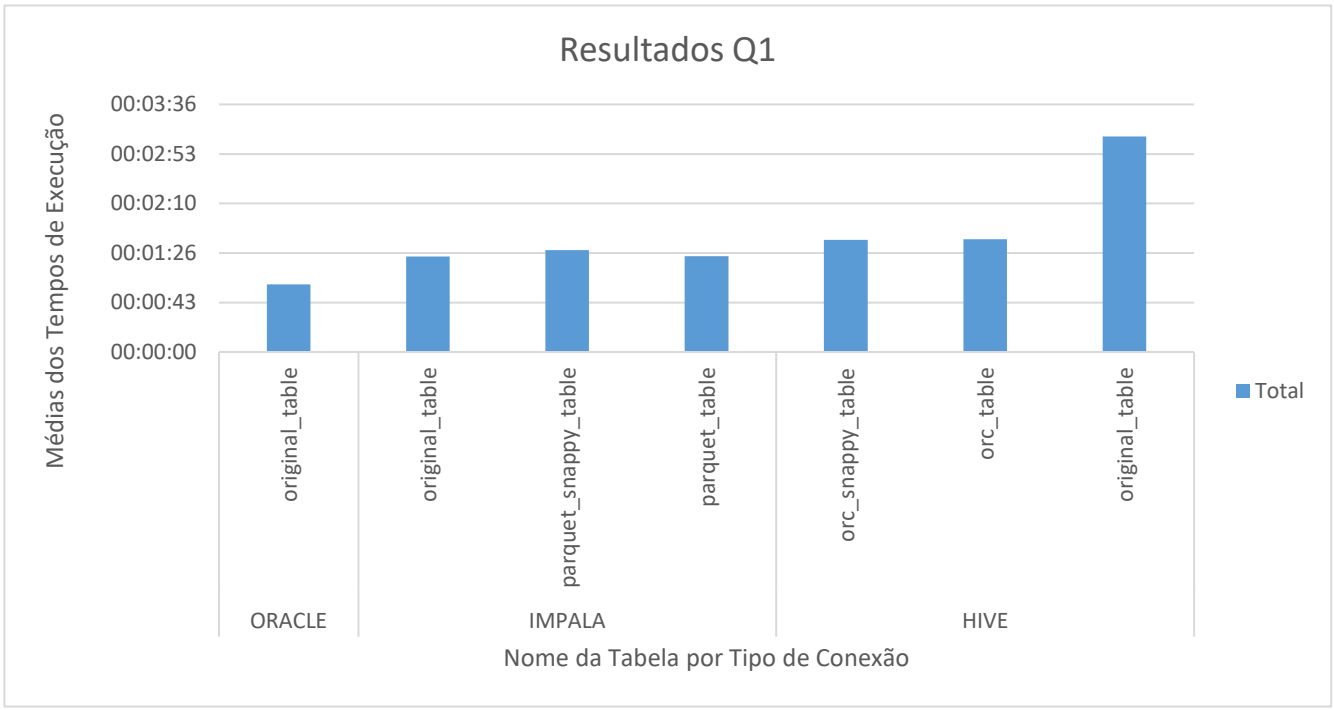


Figura 7.1 - Resultados Q1

7.3.2. Querie 2

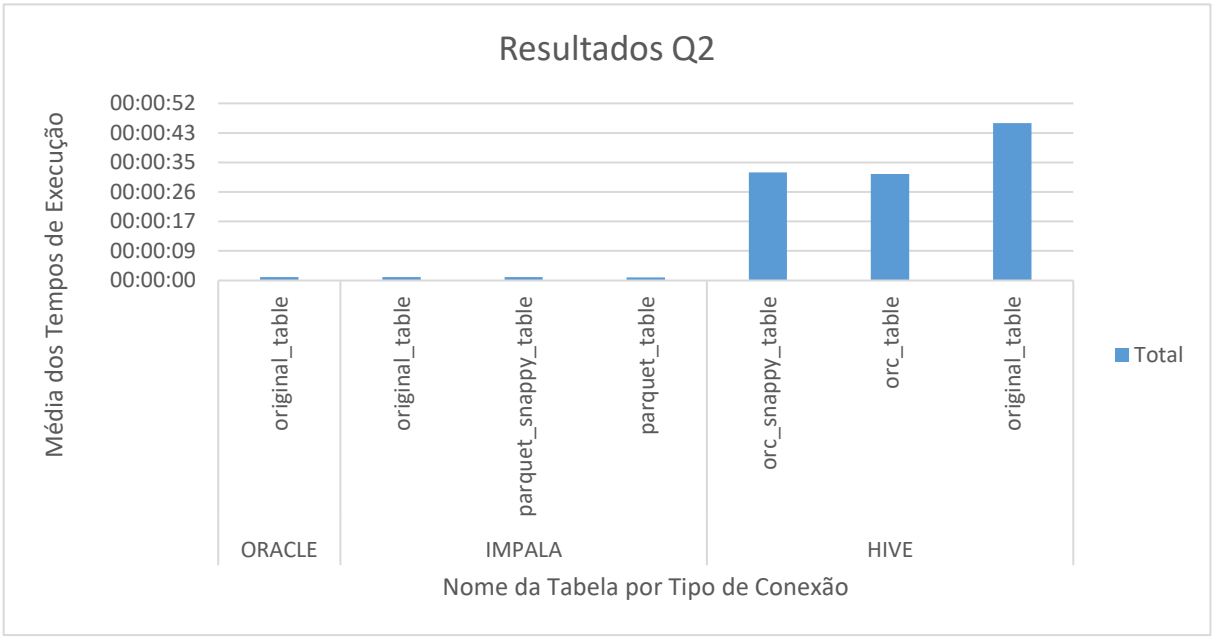


Figura 7.2 – Resultados Q2

7.3.3. Querie 3

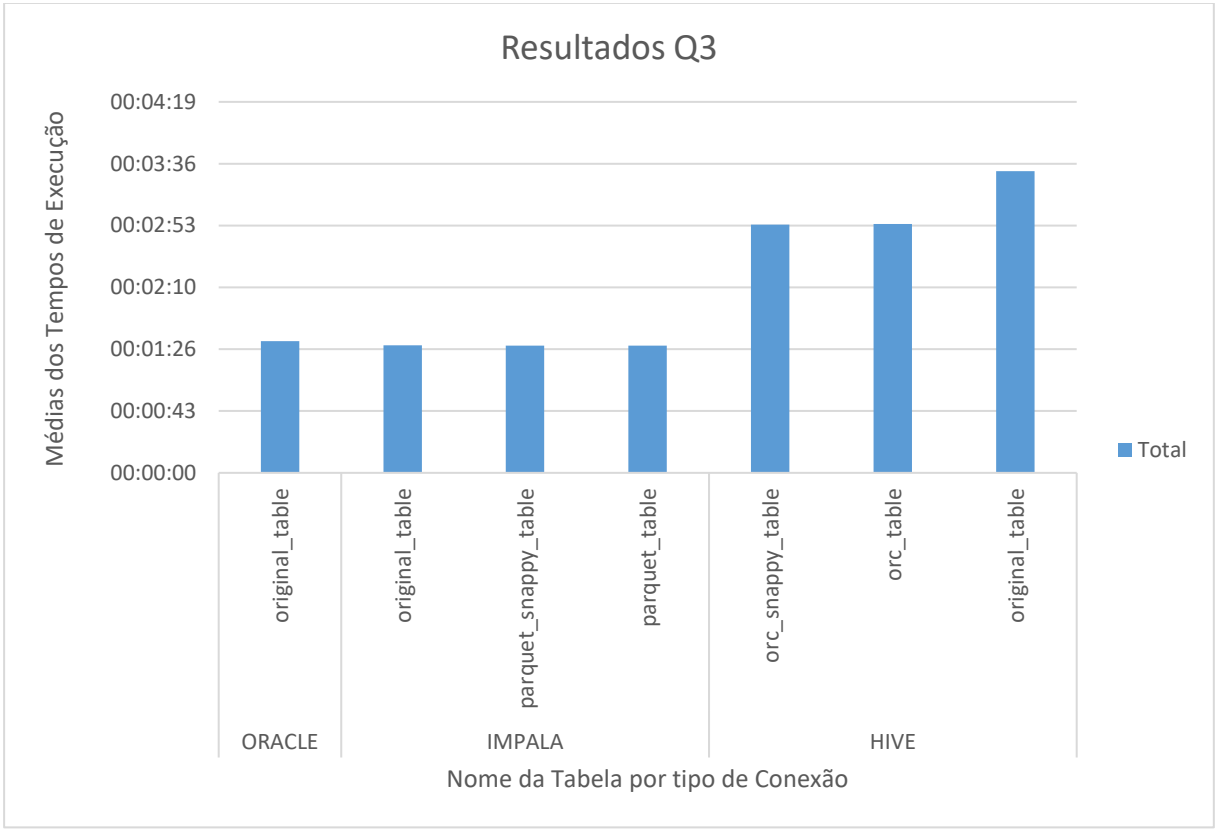


Figura 7.3 - Resultados Q3

7.3.4. Querie 4

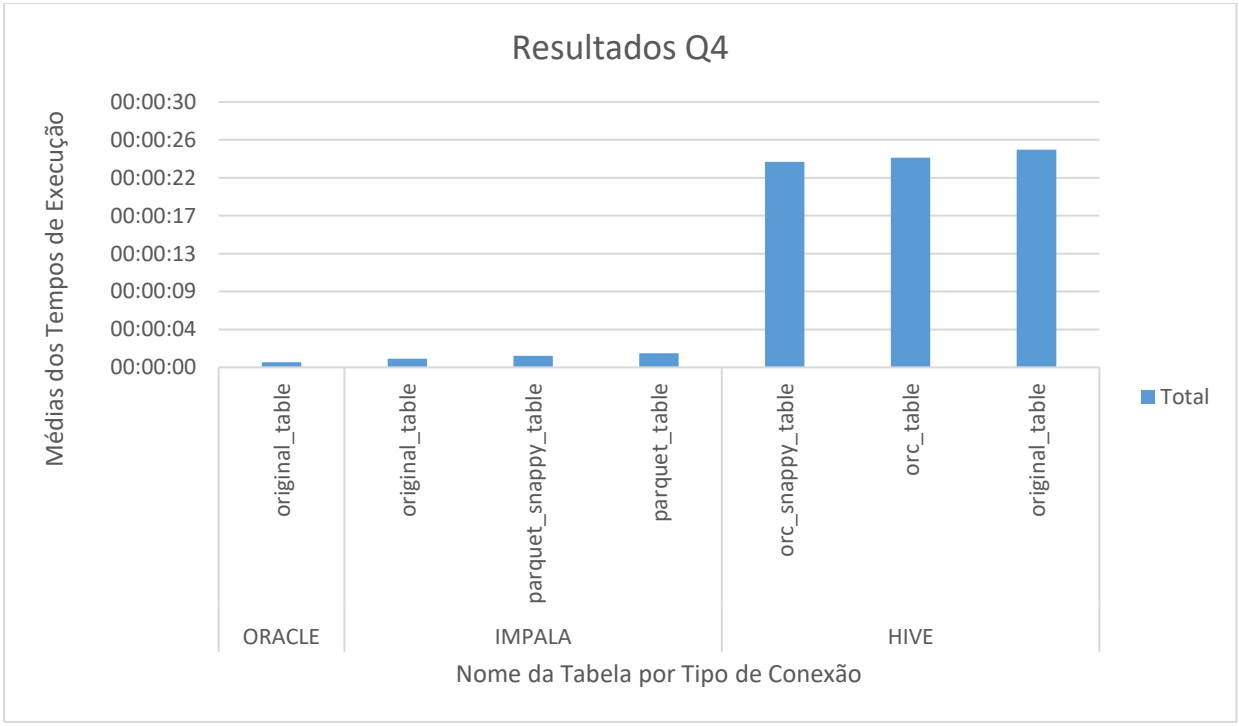


Figura 7.4 - Resultados Q4

7.3.5. Querie 5

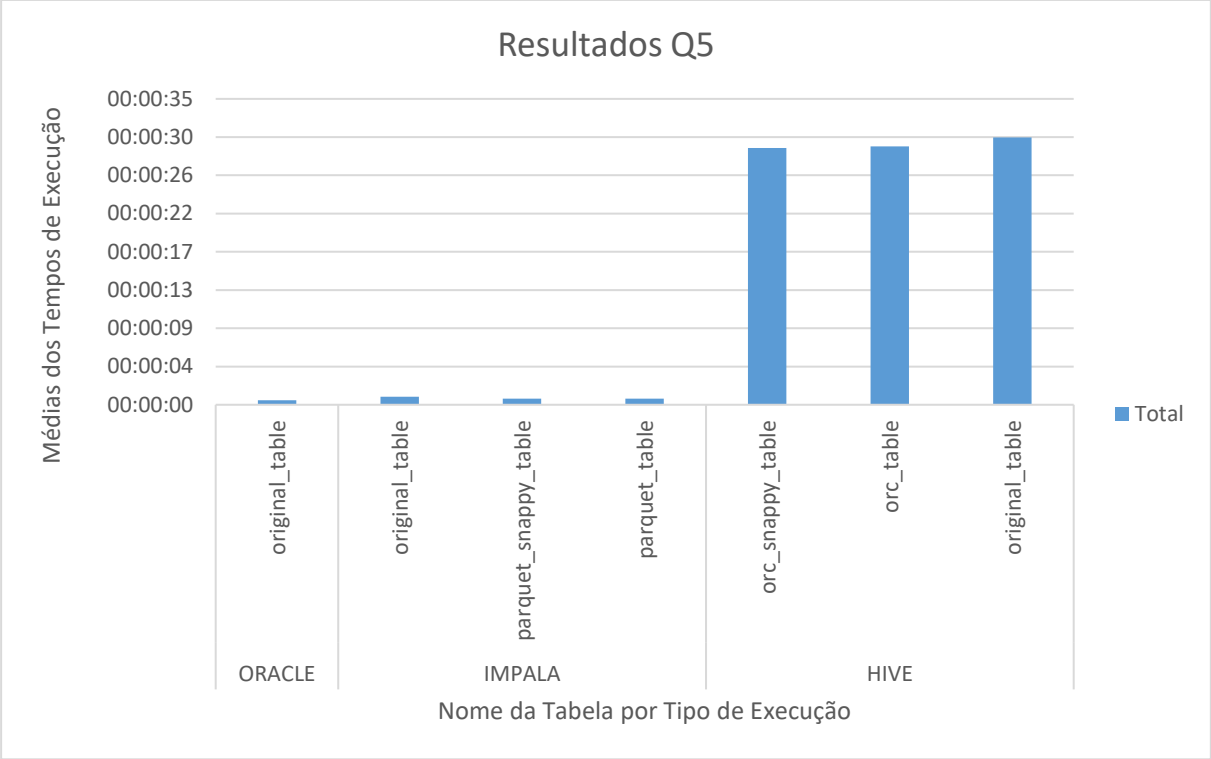


Figura 7.5 – Resultados Q5

7.3.6. Querie 6

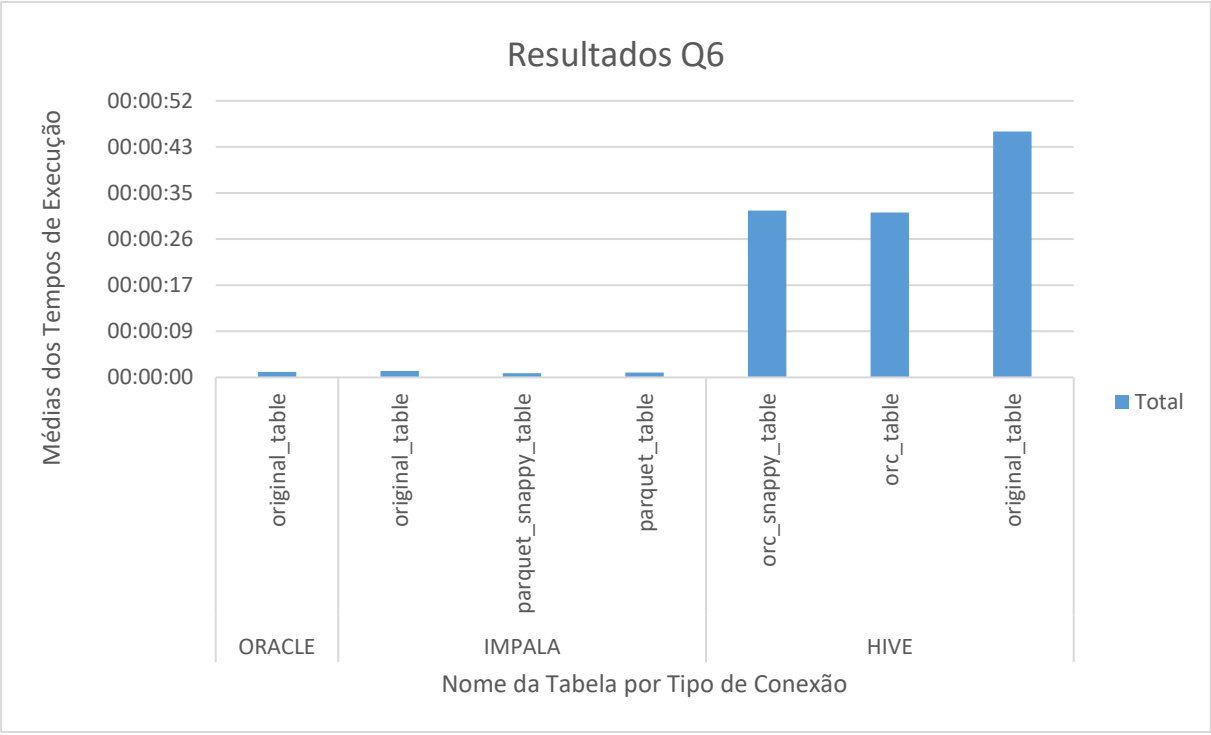


Figura 7.6 – Resultados Q6